NameSky
Security Analysis

by Pessimistic

This report is private

April 26, 2023

Abstract

In this report, we consider the security of the code base of NameSky project. Our task is to find and describe security issues in the code base of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of the code. Besides, a security audit is not investment advice.

Summary

In this report, we considered the security of NameSky project smart contracts available on a private GitHub repository. We performed our audit according to the procedure described below. The audit showed three medium-severity and several low-severity issues.

Project overview

Project description

For the audit, we were provided with NameSky project on a private GitHub repository, commit 48195a3d5adf55cd7c678aa8cfba07b8097c4032.

The scope of the audit included only the **/audit** folder.

All three tests pass successfully. The overall code coverage of the scope is 52.04%.

Procedure

We perform the audit according to the following procedure:

- **Automated analysis**
    - We compile the contracts.
    - We run provided tests and calculate code coverage using Cargo Tarpaulin.
    - We manually verify (reject or confirm) all issues reported by the following tools:
        - Cargo Geiger;
        - Cargo Audit
        - Rustle static analyzer.
- **Manual audit**
    - We manually review the code and assess its quality.
    - We check the code for known vulnerabilities.
    - We check whether the code logic complies with provided documentation.
    - We suggest possible gas and storage optimizations.
- **Report**
    - We reflect all the gathered information in the report.

Issues

We are actively looking for:

- Access control issues (incorrect admin or user identification/authorization).
- Lost/stolen assets issues (assets being stuck on the contract or sent to nowhere or to a wrong account).
- DoS due to logical issues (deadlock, state machine error, etc).
- DoS due to technical issues (Out-of-Gas error, other limitations).
- Contract interaction issues (reentrancy, insecure calls, caller assumptions).
- Arithmetic issues (overflow, underflow, rounding issues).
- Incorrect Near SDK usage.
- Other issues.

Automated analysis

Automated analysis shows the following:

- **Auto-tests**
  All three tests pass successfully. See the results in Appendix A.
- **Tests coverage**
  For code coverage calculation, we used Cargo Tarpaulin.
  The code coverage is 52.04%, which is a medium level. See the details in Appendix A.
- **Check for unsafe Rust code**
  For unsafe Rust code test, we used Cargo Geiger. The tool did not reveal any unsafe code in the core contract code. It showed minor unsafe code in the controller contract, but it was handled properly. See details in Appendix A.
- **Check for crates vulnerabilities**
  To test the codebase for crates vulnerabilities, we used Cargo Audit. The tool showed two issues, both related to near-sdk:
  - `chrono 0.4.19` recommended upgrade to `>=0.4.20`.
  - `time 0.1.43` recommended upgrade to `>=0.2.23`.

  See details in Appendix A.
- **Static analysis**
  For static analysis, we used Rustle. The tool showed 9 high-severity, 21 medium-severity, and 11 low-severity issues. However, they all were false positive.

Manual analysis

We have audited the custom NFT release and the NameSky project controller code.

The purpose of the project is to wrap and unwrap NEAR accounts in NFT for further sale or transfer.

Part of the project and auditing is a controller code that is loaded into the account wrapped in NFT and removed after unwrapping.

An external robot/service is used in the process of wrapping the account in NFT. Auditing the robot code is out of the work scope. The auditing assumed that the owner of the contract would be an ordinary user, not a DAO.

The audit showed three medium-severity and several low-severity issues.

High-severity issues

High-severity issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

The audit showed no issues of high severity.

Medium-severity issues

Medium-severity issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

M01. Possible NFT ownership manipulation

In **core/src/namesky_non_fungible_token/namesky_core_impl.rs**, users can execute the `nft_register` function declared at line 213 using a functional key (i.e., without a Full Access Key). Therefore, when a user attempts to create an NFT and performs the `nft_register` call, an attacker might intercept the functional key from this user (e.g., using an XSS attack). Having the key, the attacker can perform another `nft_register` call with this key and change the owner of the NFT before minting. Consider adding the `assert_one_yocto` check to this function.

M02. No fee withdrawal functionality

The **core** contract does not contain any functionality for `mint_fee` withdrawal. Thus, if the project considers no Full Access Keys for the contract (e.g., when the contract is conrolled by the DAO), there will be no way to withdraw fees that users paid for NFT registration.

M03. NEP-199 standard violation

In **core/src/payout_impl.rs**, the `nft_transfer_payout` function declared at line 13 does not comply with NEP-199.

In version `2.1.0` of the standard, an `Optional` argument `memo` was added to the `nft_transfer_payout` function. Also, the standard expects `approval_id` and `max_len_payout` arguments to be `Optional`.

As a result, marketplaces that follow the standard will be unable to transfer NFTs.

Low-severity issues

Low-severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

L01. No pause during the controller upgrade

The controller allows upgrading its code or rolling back. The system prevents upgrading the code to an older version by verifying that the codebase hash is unique (**core/src/namesky_non_fungible_token/controller_code.rs**, line 74, `contains_code` check). Instead, it allows rolling the code back to a previous version using the `revert_latest_controller_code` function in **src/admin_impl.rs**. If an earlier version is required, the contract must undergo consequential rollback steps. However, these steps might include a version that is undesirable for exposure on production, e.g., contains known bugs, etc. We recommend pausing the contract operation until the upgrade/rollback is complete.

L02. Inaccessibility of old versions

In **core/src/namesky_non_fungible_token/controller_code.rs**, the `get_code_views` might fail if the controller has too many versions. Consider implementing pagination and adding `offset` and `limit` arguments to this function.

L03. Misleading function name

In the **core/src/namesky_non_fungible_token/macros.rs** file, the name of the `nft_is_registered` function supposes that the function returns a value of `Boolean` type. We recommend renaming the function.

L04. Redundant argument

In the **core/src/namesky_non_fungible_token/namesky_core_impl.rs** file, the `no_access_keys` argument of the `nft_mint` function is only checked to be `true`. Consider performing this check outside the function to improve code readability.

L05. Changing ownership to a non-verified account

In **core/src/admin_impl.rs**, the `set_owner` function does not verify if the provided account name is correct. Thus, a typo in the name can result in the loss of control over a given account. We recommend changing the account owner with `set` and `approve` functions so that the new owner must confirm the gain of access.

L06. Roles separation

The owner of the **core** contract can receive fees. Since fee collection is a regular process, we recommend managing it with a separate role to ensure better protection of the owner keys.

L07. Unused export

The **controllers/controller_micro/src/contract.rs** contract exports `change_owner_id` functionality at line 26. By design, only the **core** contract can call Controller. However, it never calls this particular function.

Notes

N01. ABI is not exported

The **core** contract does not export ABI. We recommend adding ABI export to the **core** contract to simplify dependencies development.

Appendix A

Auto-tests results

```
running 3 tests
test
namesky_non_fungible_token::namesky_core_impl::test::test_register_unregister
... ok
test payout_impl::test::test_payout ... ok
test
namesky_non_fungible_token::namesky_core_impl::test::test_register_mint_redeem
... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out;
finished in 0.00s
```

Code coverage

```
cargo_tarpaulin::report: Coverage Results:
|| Uncovered Lines:
|| contracts/controllers/controller_micro/src/env.rs: 67-71
|| contracts/controllers/controller_micro/src/utils.rs: 4-5
|| contracts/core/src/admin_impl.rs: 15-16, 19-21, 24-26, 29-32, 35-36, 39-
41, 62, 66-68, 71-72, 75-77, 80-83, 86-89, 92-94, 97-99, 106-107, 114-117,
119-120, 124-126, 128-129, 137-139, 142-144, 151, 157-158, 161-162, 165-166,
169-170, 173-174, 177-179, 182-184, 187, 192-193
|| contracts/core/src/lib.rs: 55-56, 60-62
|| contracts/core/src/namesky_non_fungible_token/contract_state.rs: 25-26,
29-30, 33-34
|| contracts/core/src/namesky_non_fungible_token/controller_code.rs: 51-53,
78-79, 82-84, 100-101, 103, 105-106, 111-116, 121, 127, 130-131, 134-135,
138-139, 142-143, 146-147, 150-151, 154-155, 160-162, 165-168
|| contracts/core/src/namesky_non_fungible_token/events.rs: 36-37
|| contracts/core/src/namesky_non_fungible_token/macros.rs: 10, 54-55, 58-
59, 62, 67, 70, 76, 79, 85, 89
|| contracts/core/src/namesky_non_fungible_token/mod.rs: 77-78, 81-82, 85-
86, 89-92
|| contracts/core/src/namesky_non_fungible_token/namesky_core_impl.rs: 107,
150-151, 155-158, 163-165, 167-170, 176, 182-183, 185-188, 194, 200, 203-
206, 224-225, 227-228, 230-231, 245, 294-297, 313, 385-387
|| contracts/core/src/namesky_non_fungible_token/namesky_resolver_impl.rs:
29, 59, 74
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_approval
_impl.rs: 8, 14-16, 19-22, 25-28, 31, 37
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_core
_impl.rs: 9, 16-19, 22, 30-33
```

```
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token
_enumeration_impl.rs: 9-10, 13-14, 17-18, 21, 27
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_resolver
_impl.rs: 9, 16
|| contracts/core/src/namesky_non_fungible_token/registration.rs: 48, 54-56
|| contracts/core/src/namesky_non_fungible_token/supported_token_id.rs: 34,
36-37, 48-49
|| contracts/core/src/namesky_non_fungible_token/token_state.rs: 31, 46-48,
56-57, 64-65, 68-69
|| contracts/core/src/payout_impl.rs: 13, 21-22
|| contracts/core/src/royalty.rs: 35-36, 39-41
|| contracts/core/src/upgrade.rs: 11-13, 16-21, 23-24
|| contracts/core/src/utils.rs: 6-7
|| Tested/Total Lines:
|| contracts/controllers/controller_micro/src/env.rs: 0/5 +0.00%
|| contracts/controllers/controller_micro/src/utils.rs: 0/2 +0.00%
|| contracts/core/src/admin_impl.rs: 8/87 +0.00%
|| contracts/core/src/lib.rs: 4/9 +0.00%
|| contracts/core/src/namesky_non_fungible_token/contract_state.rs: 4/10
+0.00%
|| contracts/core/src/namesky_non_fungible_token/controller_code.rs: 25/67
+0.00%
|| contracts/core/src/namesky_non_fungible_token/events.rs: 6/8 +0.00%
|| contracts/core/src/namesky_non_fungible_token/macros.rs: 24/36 +0.00%
|| contracts/core/src/namesky_non_fungible_token/mod.rs: 9/19 +0.00%
|| contracts/core/src/namesky_non_fungible_token/namesky_core_impl.rs:
124/166 +0.00%
|| contracts/core/src/namesky_non_fungible_token/namesky_resolver_impl.rs:
25/28 +0.00%
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_approval
_impl.rs: 0/14 +0.00%
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_core
_impl.rs: 2/12 +0.00%
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_enumerat
ion_impl.rs: 0/8 +0.00%
|| contracts/core/src/namesky_non_fungible_token/non_fungible_token_resolver
_impl.rs: 0/2 +0.00%
|| contracts/core/src/namesky_non_fungible_token/registration.rs: 21/25
+0.00%
|| contracts/core/src/namesky_non_fungible_token/supported_token_id.rs: 8/13
+0.00%
|| contracts/core/src/namesky_non_fungible_token/token_state.rs: 13/23
+0.00%
|| contracts/core/src/payout_impl.rs: 8/11 +0.00%
|| contracts/core/src/royalty.rs: 21/26 +0.00%
|| contracts/core/src/upgrade.rs: 0/11 +0.00%
|| contracts/core/src/utils.rs: 4/6 +0.00%
||
52.04% coverage, 306/588 lines covered, +0.00% change in coverage
```

Cargo Geiger results

Core

```
Functions  Expressions  Impls  Traits  Methods  Dependency

0/0        0/0          0/0    0/0     0/0      ?  namesky_core 1.0.0
0/0        0/0          0/0    0/0     0/0      ?  near-contract-standards
4.1.1
57/57      502/506      0/0    0/0     0/0      !  near-sdk 4.1.1
0/0        1/1          0/0    0/0     0/0      !  near-crypto 0.14.0
0/0        1/1          0/0    0/0     0/0      !  near-crypto 0.14.0
1/1        1/1          0/0    0/0     0/0      !  near-sys 0.2.0
0/0        9/9          0/0    0/0     0/0      !  near-vm-logic 0.14.0
0/0        1/1          0/0    0/0     0/0      !  near-crypto 0.14.0
57/57      502/506      0/0    0/0     0/0      !  near-sdk 4.1.1
```

Controller

```
Functions  Expressions  Impls  Traits  Methods  Dependency

9/9        219/219      0/0    0/0     0/0      !  namesky_controller_micro
0.1.0
1/1        1/1          0/0    0/0     0/0      !  near-sys 0.2.0
```

Cargo Audit results

```
Crate:     chrono
Version:   0.4.19
Title:     Potential segfault in `localtime_r` invocations
Date:      2020-11-10
ID:        RUSTSEC-2020-0159
URL:       https://rustsec.org/advisories/RUSTSEC-2020-0159
Solution:  Upgrade to >=0.4.20
Dependency tree:
chrono 0.4.19
└── near-primitives 0.14.0
    ├── near-vm-logic 0.14.0
    │   └── near-sdk 4.1.1
    │       ├── near-contract-standards 4.1.1
    │       │   └── namesky_core 1.0.0
    │       └── namesky_core 1.0.0
    └── near-sdk 4.1.1
```

```
Crate:     time
Version:   0.1.43
Title:     Potential segfault in the time crate
Date:      2020-11-18
ID:        RUSTSEC-2020-0071
URL:       https://rustsec.org/advisories/RUSTSEC-2020-0071
Severity:  6.2 (medium)
Solution:  Upgrade to >=0.2.23
Dependency tree:
time 0.1.43
└── chrono 0.4.19
    └── near-primitives 0.14.0
        ├── near-vm-logic 0.14.0
        │   └── near-sdk 4.1.1
        │       ├── near-contract-standards 4.1.1
        │       │   └── namesky_core 1.0.0
        │       └── namesky_core 1.0.0
        └── near-sdk 4.1.1

Crate:     wee_alloc
Version:   0.4.5
Warning:   unmaintained
Title:     wee_alloc is Unmaintained
Date:      2022-05-11
ID:        RUSTSEC-2022-0054
URL:       https://rustsec.org/advisories/RUSTSEC-2022-0054
Dependency tree:
wee_alloc 0.4.5
└── near-sdk 4.1.1
    ├── near-contract-standards 4.1.1
    │   └── namesky_core 1.0.0
    └── namesky_core 1.0.0

Crate:     cpufeatures
Version:   0.2.2
Warning:   yanked
Dependency tree:
cpufeatures 0.2.2
├── sha2 0.10.2
│   ├── near-vm-logic 0.14.0
│   │   └── near-sdk 4.1.1
│   │       ├── near-contract-standards 4.1.1
│   │       │   └── namesky_core 1.0.0
│   │       └── namesky_core 1.0.0
│   └── near-primitives-core 0.14.0
│       ├── near-vm-logic 0.14.0
│       ├── near-sdk 4.1.1
│       └── near-primitives 0.14.0
│           ├── near-vm-logic 0.14.0
│           └── near-sdk 4.1.1
└── sha2 0.9.9
    └── ed25519-dalek 1.0.1
        └── near-crypto 0.14.0
            ├── near-vm-logic 0.14.0
            ├── near-sdk 4.1.1
            └── near-primitives 0.14.0
```

```
Crate:     ed25519
Version:   1.4.1
Warning:   yanked
Dependency tree:
ed25519 1.4.1
└── ed25519-dalek 1.0.1
    └── near-crypto 0.14.0
        ├── near-vm-logic 0.14.0
        │   └── near-sdk 4.1.1
        │       ├── near-contract-standards 4.1.1
        │       │   └── namesky_core 1.0.0
        │       └── namesky_core 1.0.0
        ├── near-sdk 4.1.1
        └── near-primitives 0.14.0
            ├── near-vm-logic 0.14.0
            └── near-sdk 4.1.1

error: 2 vulnerabilities found!
warning: 3 allowed warnings found
```

This analysis was performed by Pessimistic:

Sergey Grigoriev, Security Engineer
Nikita Kuznetsov, Security Engineer
Evgeny Marchenko, Senior Security Engineer
Boris Nikashin, Analyst
Irina Vikhareva, Project Manager
Alexander Seleznev, Founder

April 26, 2023