AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

TABLE OF CONTENTS

contact@auditmos.com

AUDITMOS

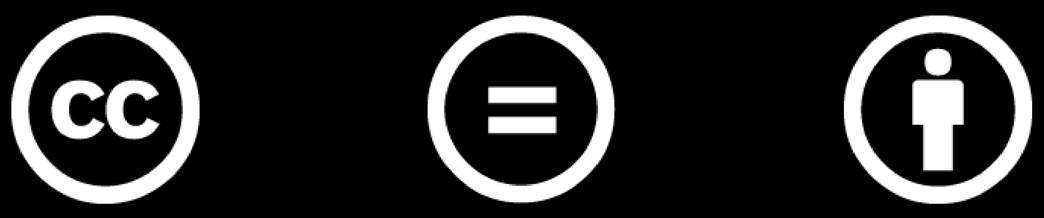
Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

LICENSE



contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

DISCLAIMER

contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

APPROACH AND METHODOLOGY

PURPOSE

1. Determine the correct operation of the protocol, according to the design specification.
2. Identify possible vulnerabilities that could be exploited by an attacker.
3. Detect errors in the smart contract that could lead to unexpected behavior.
4. Analyze whether best practices were followed during development.
5. Make recommendations to improve security and code readability.

CODEBASE

Repository	https://github.com/TerrnadoCash/terrnado-contracts/
Branch	main
Commit hash	7a5dd1398deb5adb4e1a4e86914d1f5e8ad6c8d7

METHODOLOGY

1. Reading the available documentation and understanding the code.
2. Doing automated code analysis and reviewing dependencies.
3. Checking manually source code line by line for security vulnerabilities.
4. Following guildlines and recommendations.
5. Preparing this report.

contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

DESCRIPTION

Issues Categories:

Severity	Description
CRITICAL	vulnerability that can lead to loss of funds, failure to recover blocked funds, or catastrophic denial of service.
HIGH	vulnerability that can lead to incorrect contract state or unpredictable operation of the contract.
MEDIUM	failure to adhere to best practices, incorrect usage of primitives, without major impact on security.
LOW	recommendations or potential optimizations which can lead to better user experience or readability.

Each issue can be in the following state:

State	Description
PENDING	still waiting for resolving
ACKNOWLEDGED	know but not planned to resolve for some reasons
RESOLVED	fixed and deployed

contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

AUDIT SCOPE

1. getting to know the project ✓
2. research into architecture ✓
3. manual code read ✓
4. check of permissions ✓
5. identify common Rust vulnerabilities ✓
6. test coverage ✓
7. static analysis ✓

contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

FINDINGS

Finding	Severity	Status
#1 - improve tests code coverage	LOW	RESOLVED
#2 - catch common mistakes and improve your Rust code	LOW	RESOLVED
#3 - bigint is unmaintained, use uint instead	LOW	RESOLVED
#4 - satisfy license requirements	LOW	ACKNOWLEDGED

contact@auditmos.com

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

#1 - IMPROVE TESTS CODE COVERAGE

Severity	Status
LOW	RESOLVED

Test Coverage is an important indicator of software quality and an essential part of software maintenance. It helps in evaluating the effectiveness of testing by providing data on different coverage items. It is a useful tool for finding untested parts of a code base. Test coverage is also called code coverage in certain cases.

Test coverage can help in monitoring the quality of testing and assist in directing the test generators to create test cases that cover areas that have not been tested. It helps in determining a quantitative measure of Test coverage, which is an indirect measure of quality and identifies redundant test cases that do not increase coverage.

RECOMMENDATION

It is highly recommended to test all of the functions and have high ratio of test coverage. It is recommended to use code coverage reporting tool for the Cargo build system for example cargo-tarpaulin.

```
|| Uncovered Lines:
|| contracts/terrnado-anonymizer/src/contract.rs: 53, 69-70
|| contracts/terrnado-anonymizer/src/querier.rs: 3, 7-9
|| contracts/terrnado-anonymizer/src/tools.rs: 15, 19, 21-24, 26, 34
|| Tested/Total Lines:
|| contracts/terrnado-anonymizer/src/contract.rs: 22/25
|| contracts/terrnado-anonymizer/src/execute.rs: 64/64
|| contracts/terrnado-anonymizer/src/querier.rs: 0/4
|| contracts/terrnado-anonymizer/src/queries.rs: 14/14
|| contracts/terrnado-anonymizer/src/tools.rs: 9/17
||
87.90% coverage, 109/124 lines covered
```

PROOF OF SOURCE

https://github.com/TerrnadoCash/terrnado-contracts/tree/7a5dd1398deb5adb4e1a4e86914d1f5e8ad6c8d7/contracts/terrnado-anonymizer/src/testing

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

#2- CATCH COMMON MISTAKES AND IMPROVE YOUR RUST CODE

Severity	Status
LOW	RESOLVED

Using cargo clippy we found a collection of lints with common mistakes. Please consider them and improve your Rust code.

RECOMMENDATION

1) redundant clone

contracts/terrnado-anonymizer/src/execute.rs:71:36

Please remove clone() from

to_address: proposal.to.clone().to_string()

2) using clone on type cosmwasm_std::Uint128 which implements the Copy trait

contracts/terrnado-anonymizer/src/execute.rs:73:25

Please remove clone() from

proposal.amount.clone()

3) writing &String instead of &str involves a new object where a slice will do

contracts/terrnado-anonymizer/src/querier.rs:5:20

Change this contract_addr: &String to contract_addr: &str

Change contract_addr.clone() to contract_addr.to_string()

4) unneeded return statement

contracts/terrnado-anonymizer/src/tools.rs:12:5

Remove return Err(ContractError::Unauthorized {});

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

#2- CATCH COMMON MISTAKES AND IMPROVE YOUR RUST CODE

Severity	Status
LOW	RESOLVED

Using cargo clippy we found a collection of lints with common mistakes. Please consider them and improve your Rust code.

RECOMMENDATION

5) writing `&Vec<_>` instead of `&[_]` involves one more reference and cannot be used with non-Vec-based slices

contracts/terrnado-anonymizer/src/tools.rs:17:26

Change this `contracts_addresses: &Vec<String>` to `contracts_addresses: &[String]`

6) this expression borrows a reference `(&std::string::String)` that is immediately dereferenced by the compiler

contracts/terrnado-anonymizer/src/tools.rs:22:52

Change this `let addr_form = deps.api.addr_validate(&contract);`
to `let addr_form = deps.api.addr_validate(contract);`

7) returning an `Err(_)` with the `?` operator

contracts/terrnado-anonymizer/src/tools.rs:24:17

Instead of
`Err::<(), StdError>(StdError::generic_err("Cannot convert address to addr"))?;`
use
`return Err(StdError::generic_err("Cannot convert address to addr"))`

8) this creates an owned instance just for comparison

contracts/terrnado-anonymizer/src/tools.rs:32:49

Instead of `"uusd".to_string()` use `*"uusd"`

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

#3- BIGINT IS UNMAINTAINED, USE UINT INSTEAD

Using cargo audit we found a use of not maintained library.

Severity	Status
LOW	RESOLVED

RECOMMENDATION

```
Crate:         bigint
Version:       4.4.3
Warning:       unmaintained
Title:         bigint is unmaintained, use uint instead
Date:          2020-05-07
ID:            RUSTSEC-2020-0025
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0025
Dependency tree:
bigint 4.4.3
└── cosmwasm-bignumber 2.2.0
    └── terrnado-anonymizer 1.0.0
```

AUDITMOS

Secure your space

TERRNADO (ANONYMIZER)

audit / code review report

February 10, 2022

#4- SATISFY LICENSE REQUIREMENTS

Severity	Status
LOW	ACKNOWLEDGED

Using cargo deny check we found that you do not follow the requirements regarding to license from your dependencies.

RECOMMENDATION

One important aspect that one must always keep in mind when using code from other people is what the licensing of that code is and whether it fits the requirements of your project. Luckily, most of the crates in the Rust ecosystem tend to follow the example set forth by Rust itself, namely dual-license MIT and Apache 2.0, but of course, that is not always the case.

contact@auditmos.com

auditmos.com

Secure your space

contact@auditmos.com