SecurityDAO

DAO DAO Audit

DAO DAO CosmWasm Contract Audit

Prepared by: Logan Cerkovnik
Date: May 5th, 2022

TVL : **$>=10M USD equivalent**

... + more about threat model

Outline

Document Revision History

Version	Modification	Date	Author
0.1	Created	3/19/2022	Logan Cerkovnik
0.2	Updated	4/20/2022	Logan Cerkovnik
0.3	Updated	5/6/2022	Logan Cerkovnik

Contact

Contact	Organization	Email
Logan Cerkovnik	Security DAO	logan@secdao.xyz
Paul Wagner	Security DAO	paul@secdao.xyz
Barton Rhodes	Security DAO, DAODAO	barton@secdao.xyz

Executive Overview

Audit Summary

Security Dao worked on an engagement with DAO DAO from 3/14/2022 through 4/08/2022 to conduct a security assessment of DAO DAO contracts to enable major breaking changes and improvements for DAO DAO version 1.0 upgrade.

The security engineers involved with the audit are security and blockchain smart contract security experts with advanced knowledge of smart contract exploits.

The purpose of this audit is to achieve the following:

- **Ensure** that smart contract functions work as intended
- **Identify** potential security issues with the smart contracts

In summary, Security Dao identified impactful improvements to reduce the likelihood and scope of risks, which were addressed by the WasmSwap team.

The **primary ones** are as follows:

- Use of yanked dependency (from cosmwasmw-std sub-dependency)

External threats such as intercontract functions and calls should be validated for expected logic and state and are not covered within the scope of this audit. Only direct rpc contract interaction is considered here not any UI components or frontend wasm interactions are excluded.

Test Approach and Methodology

Security DAO performed a combination of manual review of the code and automated security testing.

The following phases were used throughout the audit:

- Research into the architecture, purpose, and use of the platform
- Manual code review and walkthrough
- Manual Assessment of the use and safety for critical rust variables and functions in scope to identify any contracts logic related vulnerability
- Fuzz Testing (securitydao fuzzing tool)
- Check Test Coverage (cargo tarpaulin)

  95.76% coverage, 6999/7309 lines covered,

- Scanning of Rust files for vulnerabilities (cargo audit)

Dependency	Version	Warning
const-oid	0.6.0	yanked
crypto-bigint	0.2.2	yanked

Risk Methodology

Risk Likelihood and impact scales 1 through 5 where 5 is the most severe

Risk Likelihood Scale

1 low	2 unlikely	3 possible	4 likely to happen	5 high

`least severe` `most severe`

A low likelihood risk indicates that the likelihood of attack is low because of obscurity or requiring additional exploits to utilize, a possible attack is one that is possible but not an attack method commonly seen in the wild or well-known, and high risk likelihood represents an exploit extremely likely to be used, readily apparent, or commonly been used in the past against similar systems

Risk Impact Scale

1 low	2 limited	3 Impactful	4 critical	5 severe

least severe most severe

In the context of smart contracts, a low risk impact might be something associated with limited scope or a preventive best practice, an impactful risk may result in large loss of funds but not in a systematic way, and a severe risk impact could result in substantial loss of funds in a systematic way.

Scope

Cosmwasm Smart Contracts

The primary target for the audit is cw-governance, voting, and dao modules. User interface and cross contract messages are considered out of scope for this work.

- **Repo:** Public: https://github.com/DA0-DA0/dao-contracts/tree/zeke/contracts-v1
- **Commit hash: bceff1805c9ccea12124fac0d15e7d4967b76c5e**
- **Updated Commit hash: 41898e17ab4f8f4d67b7217927656a3623661dd9**

Action Plan

Low Level of Effort to Fix and High Impact

- **(SEC - 22)** Proposals Can Be Executed Multiple Times After Passing
- **(SEC - 20)** Frontend Crashes After Getting Adding CW20

Low Level of Effort to Fix and Low Impact

- **(SEC - 13)** Upgrade Yanked Dependencies
- **(SEC - 23)** Passed Proposals Can't Be Executed After Expiration

High Level of Effort to Fix and Low Impact

- **(SEC - 21)** Frontend Crashes After Getting Adding CW20

High Level of Effort to Fix and High Impact

- **(SEC - 20)** No Treasury Protections for Outbound Funds

Assessment Summary and Findings Overview

Findings and Tech Details

(SEC - 13) Yanked Dependencies

Severity Low / Impact Low

Description

Dependencies for const-oid version 0.6.0 and crypto-bigint 0.2.2 are both yanked. This is a cosmwasm-std 1.0.0beta dependency issue.

Code Location

`https://github.com/DA0-DA0/dao-contracts/blob/zeke/contracts-v1/contracts/cw4-registry/Cargo.lock`

Risk Level

The risk likelihood is low and the impact is low

Recommendation

Upgrade cosmwasm-std to latest version 1.0.0beta8

Remediation Plan

(SEC - 20) No Treasury Protections for Outbound Funds

High Severity / Medium Impact

Description

There is no ability to protect a treasury against an attack where a liquid governance token is able to be bought quickly on a dex, reach voting consensus, and then propose to remove all of the governance token value from the treasury. This sort of attack will become more prevalent with increasing usage of cosmos ecosystem lending protocols and more DAO's listing CW20 tokens on dexes.

The way this can be prevented is:

- Add the ability to restrict or approve transfer of token/ trading on amm transfers in DAO DAO contracts
- Add max treasury liquidation parameter in a single proposal
- Increasing voting requirements based on size of transaction for outbound treasury transactions
- Restrict the ability of a proposal to send more of the treasury than voting participants or yes vote fraction on a proposal
- Add an educational page to the website settings for an tradable gov token Dao vs a non-tradeable gov token dao and how to setup securely

Code Location

`dao-contracts/contracts/cw-core/src/contract.rs`

Risk Level

Risk likelihood is medium and severity is high. Not every Dao will have a liquid token that could be stolen this way

Recommendation

Add some of the features proposed above to mitigate this type of attack.

Remediation Plan

(SEC - 21) Frontend Crashes on Out of Gas Error After Getting Adding CW20

Low Severity / High Impact

Description

When a dao is given or creates more than 4 cw20 tokens the user interface is disabled by an out of gas error on the js rpc query which results in denial of dao ui. This alone cannot destroy funds, but could be used in conjunction with other attacks or in a denial of service capacity

Code Location

`dao-contracts/contracts/stake-cw20/src/contract.rs`

Risk Level

Risk likelihood is high and severity is low. Every Dao is impacted, but funds are not at risk.

Recommendation

Fix UI queries so they do not break on multiple cw20 token.

Remediation Plan

(SEC - 22) Proposals Can Be Executed Multiple Times After Passing

High Likelihood / High Impact

Description

A proposal that passes can be executed multiple times before expiration. This was exploited in the v1 version of the contracts used in a DAO and discovered during an incident response investigation. This bug was originally discovered by @ezekiiel on the DAODAO team. This bug does not impact earlier versions of DAODAO contracts before version 1.0.

Code Location

https://github.com/DA0-DA0/dao-contracts/blob/main/contracts/cw-core/src/contract.rs

Risk Level

Risk likelihood is high and severity is high. Any DAO fund transfer proposal is vulnerable to being exploited this way which could drain the entire treasury maliciously.

Recommendation

Need to modify state so that proposals can only be executed a single time.

Remediation Plan

Dao's with the vulnerable proposal module will need to upgrade to a patched proposal module.

(SEC - 23) Passed Proposals Can't Be Executed After Expiration

High Likelihood / Low Impact

Description

A proposal that passes but is never executed before expiration is unable to be executed. This results in the proposal needing to be redone. This bug was also discovered by @ezekiiel from the DAODAO team.

Code Location

https://github.com/DA0-DA0/dao-contracts/blob/main/contracts/cw-core/src/contract.rs

Risk Level

Risk likelihood is high and severity is low. This will not likely impact funds , but could delay response times and governance proposals.

Recommendation

Need to modify state so that proposals can only be executed a single time but any time after passing or passed right after expiration automatically.

Remediation Plan