Ton Meme DAO: **Contract Audits**

September 18, 2025

https://t.me/coinsult_tg

info@coinsult_tg

Manual Audit Report

Code can be found on:

https://github.com/Ton-Meme-Dao/contracts

Issues on the next page

Grab Admin Rights

Right after you deploy the Registry, the admin field is empty. Your rule says "if admin is empty, anyone can set it." That means the first random caller on-chain can make themselves admin before you do and then control everything (fees, DAO deployments, pushing admin messages).

Suggested fix:

- Set the admin address during init and only let the current admin change it later.
- Remove the "if admin is empty, anyone can set it" exception.

✅ Noted: admin rights already taken by the team

Empty Contracts

ProposalDeployer.receive(DeployAndInitProposal)
ProposalDeployer.receive(SendUpdateProposal)
Dao.receive(FwdMsg) (indirectly, by forwarding whatever it's given)

Your sends are configured to forward "everything left in the contract" as value. If a call sequence leaves a lot of funds in a contract (intentionally or not), the next send can sweep it all out. A crafted forwarded message could also trick the DAO into sending away more than intended.

Suggested fix:

- Never use the "send remaining value" mode for regular logic.
- Always send a specific, small amount needed for gas and storage.
- For the DAO forwarder, reconstruct outgoing messages yourself and cap the value that can be sent out.

✅ Fixed

No forward limits

The DAO forwarder passes along the exact message it receives. It doesn't hold back any funds to stay alive, and it doesn't block dangerous flags like "send remaining value." That means the DAO could accidentally or maliciously forward too much money and become unable to pay rent.

Suggested fix:

- Before sending, reserve a safe minimum balance so the DAO can't go broke.
- Rebuild the outgoing send with your own safe parameters:
    - Cap the amount that can be forwarded.
    - Disallow "send remaining value" and other sweeping modes.
    - Prefer bounced messages so you see failures instead of silently losing them.

✅ Fixed

Over and under fund

ProposalDeployer.receive(DeployAndInitProposal)
ProposalDeployer.receive(SendUpdateProposal)

The deployer forwards whatever value is left instead of sending a fixed amount. If too little value arrives, deployment fails; if too much arrives, you accidentally push out a lot of funds.

Suggested fix:

- Always send a fixed, known-good amount that safely covers gas and storage.
- Don't forward all remaining funds.
- Use bounce so you get an error back if the destination isn't ready.

✅ Fixed

General notes:

You compute a DAO address by index and send with bounce disabled. If the DAO at that index doesn't actually exist yet, the message just disappears and you won't notice.

✅ Noted

Both owner and proposalOwner can send arbitrary internal messages "as the DAO." If the proposal owner key is hot or compromised, they could forward risky messages and drain or misconfigure things.

✅ Noted

Users can stuff very large strings into on-chain storage. While cells are bounded, this can still drive up costs, make updates expensive, or risk hitting limits.

✅ Noted

Disclaimer

This audit report has been prepared by Coinsult's experts at the request of the client. In this audit, the results of the static analysis and the manual code review will be presented. The purpose of the audit is to see if the functions work as intended, and to identify potential security issues within the smart contract.

The information in this report should be used to understand the risks associated with the smart contract. This report can be used as a guide for the development team on how the contract could possibly be improved by remediating the issues that were identified.

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.