

THE RESEARCH UNIVERSITY PRESS

International
Computational Law Journal

VOLUME 2 / ISSUE 4 / CURATED SOURCE EDITION

Exploiting Inattention and Misconfigurations in Composable Governance Systems

Isaac Patka

MIT Computational Law Report · Originally published 2022-12-05

Independent reader edition under CC BY 4.0. This attribution cover was added; original source PDF pages follow unchanged. Original authorship and source identity are retained. MIT, Stanford University, Stanford Law School and CodeX have not sponsored or endorsed this republication.

Source, license and version evidence accompany the online article.

MIT Computational Law Report

Exploiting Inattention and Misconfigurations in Composable Governance Systems

Isaac Patka

Published on: Dec 05, 2022

URL: https://law.mit.edu/pub/exploitinginattentionandmisconfigurations

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)

ABSTRACT

Many DAOs desire gasless voting solutions to make participation easier for members. Some gasless voting solutions that have been adopted by DAOs rely on easily exploited oracle systems to bring votes on chain for execution. A real world example is demonstrated where ~$20,000, or 50% of a DAOs treasury was taken through a fraudulent transaction. Other DAOs that have implemented the same system manage individual treasuries exceeding $10-20M. There are simple steps these DAOs can take to safeguard their treasuries, but the exploit highlights an issue in composable governance and treasury management where tools are adopted without thorough vetting and proper configuration.

Visit the web version of this article to view interactive content.

Work from this article was presented by the author at Devcon Bogotá in a talk titled “Exploiting Inattention & Optimism in DAOs”

Introduction

Blockchains only know information that is internal to itself. This includes a wallet’s balance of an asset for the native network currency, a wallet’s balance of a governance token issued by a DAO, the owner of an NFT, and to some extent information about itself like the current time & date, block number, and other protocol metadata. But if you ask a blockchain about something external, like the weather, a stock price, the exchange rate of two fiat currencies, or the results of an off-chain vote, it will have to rely on some trusted place to source that information. These data sources are called Oracles.

Oracles can be implemented in various ways. The most popular oracle provider is Chainlink, which supplies market data, verifiable randomness, weather, and many other data feeds to smart contracts. Markets secured by Chainlink exceed $20B in assets. In the Chainlink ecosystem, node operators are paid in LINK tokens to provide accurate real-world data. Node operators must also stake LINK to provide data. Staked LINK can be taxed or slashed if dishonest behavior is detected.

Reality.ETH is the oracle system that is the main focus of this analysis. At the time of this analysis, DAOs guarding their treasury with a Reality.ETH module have a total value exceeding $30M.

Without careful selection of an arbitrator, the Reality.ETH oracle system can easily be made to return false information as the highest bidder becomes the trusted answer on chain. Worse, the participants in the Reality.ETH Q&A may continue submitting bids past the point of profitability, as demonstrated by the Dollar Auction thought experiment/ paradox.

As DAOs adopt composable governance tooling, we need to figure out how to vet, monitor, and educate governance participants. Until then we will start to see more exploits that are more related to misconfigurations and misunderstandings, rather than pure smart contract bugs.

How it is supposed to work

Reality.ETH works via questions and answers. Some asks a question by interacting with a smart contract on Ethereum. The question can then be answered by Reality.ETH users. Answerers attach a bond to their answer that they are willing to lose if they are proven wrong. The system allows users to override a previous answer to a question by submitting a higher bond. This is supposed to incentivize the correct answer to ultimately be submitted on chain.

Askers can optionally elect an arbitrator to settle a disputed question. In naïve implementations of the system, the arbitrator is sometimes specified to be the same Gnosis Safe that has enabled the module. Other times it is skipped altogether as the user does not understand the failure that can occur. If no arbitrator is elected, the answer with the highest bond is automatically the chosen answer.

Once an answer has been selected as the truth, there is a grace period in which the Gnosis Safe signers can veto the decision. Getting multisig signers to respond to timely requests is very challenging as most DAOs do not have an ‘on call’ system where people are near their signing keys at set times.

Failure Mode

With no arbitrator, the Reality.ETH security model bears an unfortunate resemblance to the Dollar Auction game.

From Wikipedia:

> The dollar auction is a non-zero sum sequential game explored by economist Martin Shubik to illustrate a paradox brought about by traditional rational choice theory in which players are compelled to make an ultimately irrational decision based completely on a sequence of apparently rational choices made throughout the game.

A dollar bill goes up for auction with 2 rules. Highest bid wins the dollar. But the 2nd place bidder also loses their bid.

Bidding starts and A places a bid for $0.25, seeing a seemingly rational opportunity to make a quick $0.75 profit. B places a counter bid for $0.35, and so on. Bidding continues and nears $1. Both A & B realize that if they win, they break even, and if they lose they are out $1. At this point both sides try to minimize losses by bidding over $1. This continues until one side is out of money.

What if we used Reality.ETH to distribute the winnings…

The winner of the dollar auction is instructed to claim their winnings from smart contract that uses Reality.ETH to determine the winner. The question ‘who won the dollar auction’ is asked. The winner submits an answer that they won, placing $1 as a bond. The loser sees an opportunity to recoup their losses from the auction and places a $2 bond claiming they won, receiving the $1 bond from the true winner in the process. Much like the dollar auction, this process continues until one side is out of money. There is no guarantee that the answer brought on chain is true, and both sides probably lost even more money in the process.

Other flaws

Lack of attention

Reality.ETH relies on the human actors in the system actually paying attention. However even though $30M+ is secured by Reality.ETH voting modules, many do not have notification infrastructure set up to see when questions are asked.

Capital requirements

Reality.ETH may be harder to exploit in a scenario where both altruistic and malicious market participants have unlimited capital and attention. However, the questions being asked of Reality.ETH often control sums of money that outweighs the balances available in the wallets of the answers. If a highly capitalized malicious actor decides to answer a question incorrectly, they will likely succeed by being able to outspend their adversary.

Arbitration misconfigurations

Reality.ETH allows for an arbitrator to be selected for a question template. When an arbitrator is not selected, the system defaults to a ‘highest bidder wins’ scenario, i.e. the dollar auction.

Some DAOs have also made the mistake of electing themselves as the arbitrator. This is particularly risky because if the signers lose access to their keys, either permanently or temporarily, the system behaves like a no-arbitrator configuration.

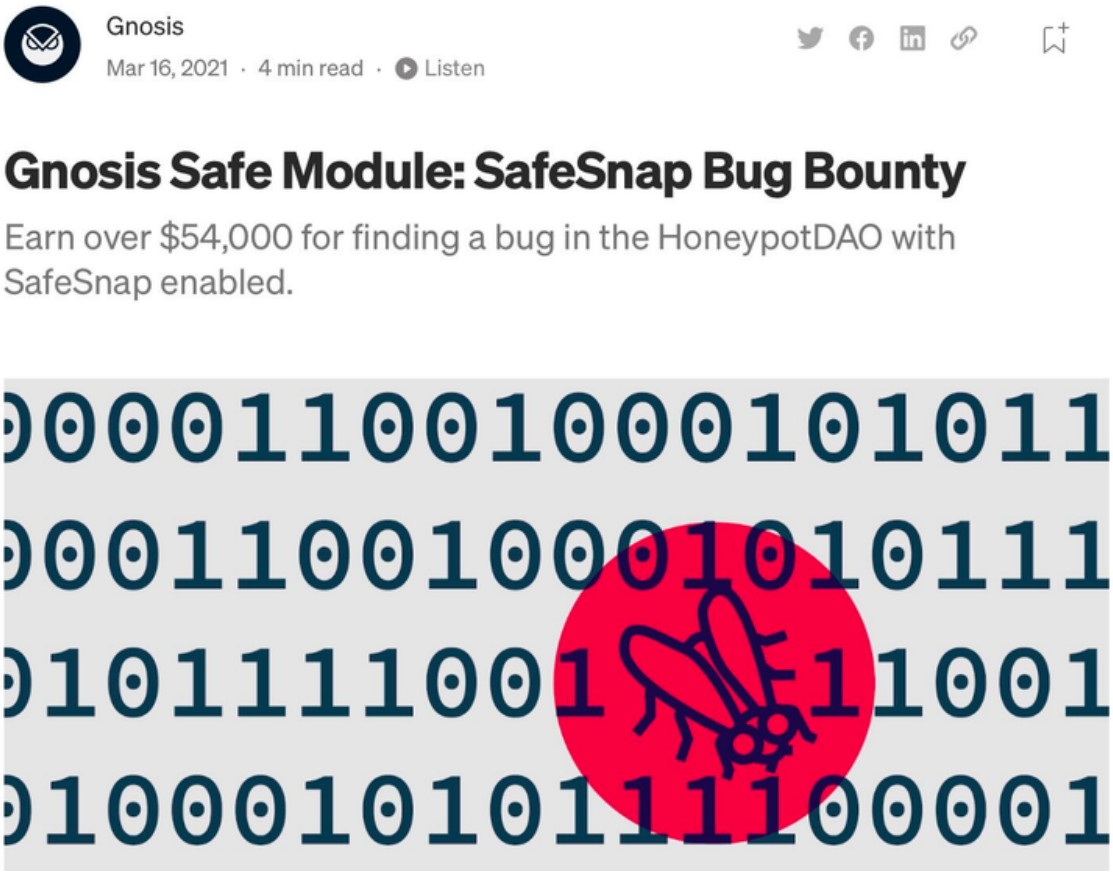
The lesson of a dollar auction is that the best way to win is to never join one. Perhaps the lesson here is that the best way to ask Reality.ETH a question is to never ask one.

If it is so flawed, why is it gaining adoption?

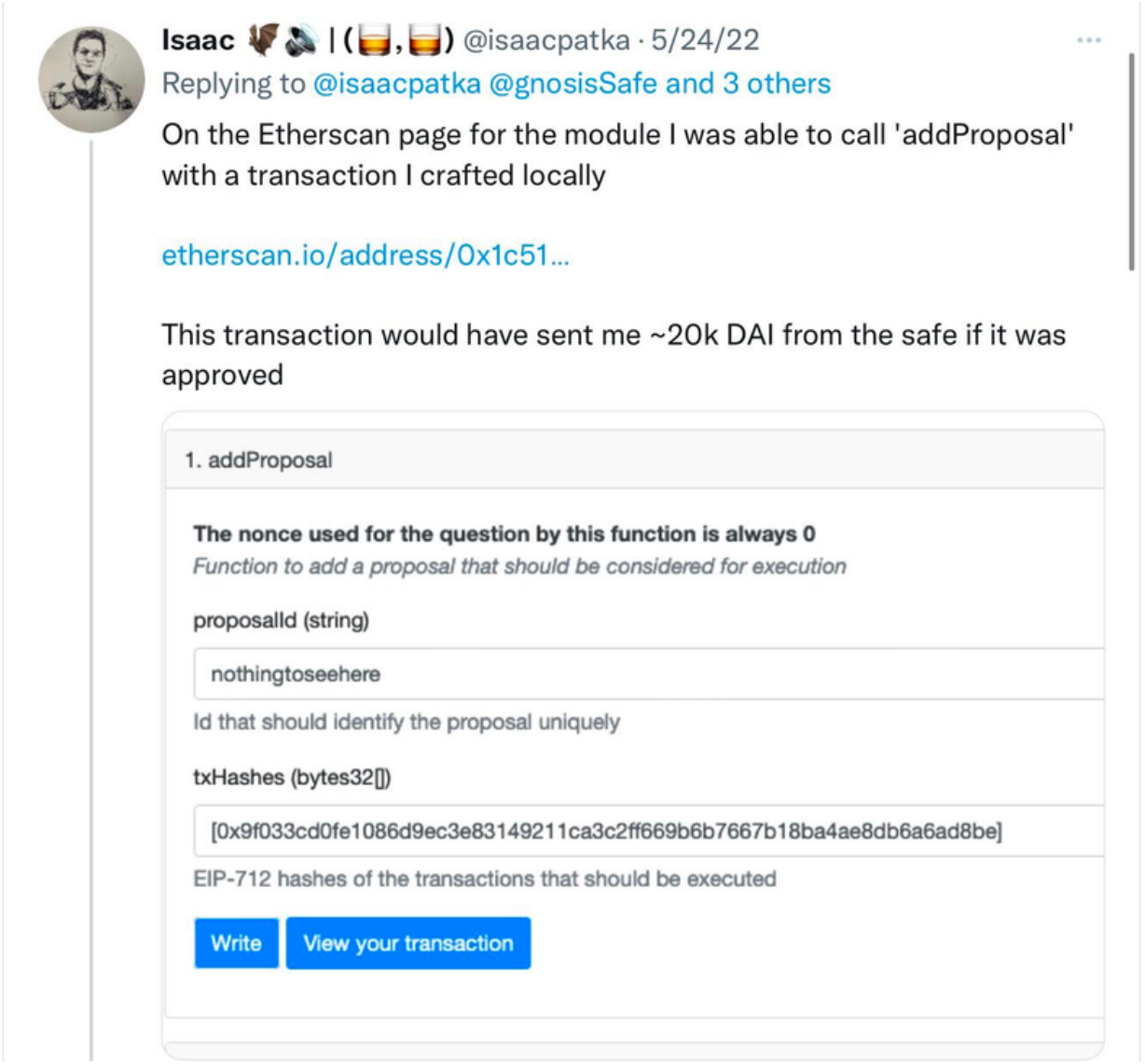
Composable governance systems must be careful about what modules they allow users to easily add onto their organizations and treasuries. When the expectations of users for curation and vetting does not match the rigor of analysis by platforms and interfaces, it leaves systems vulnerable to exploitation.

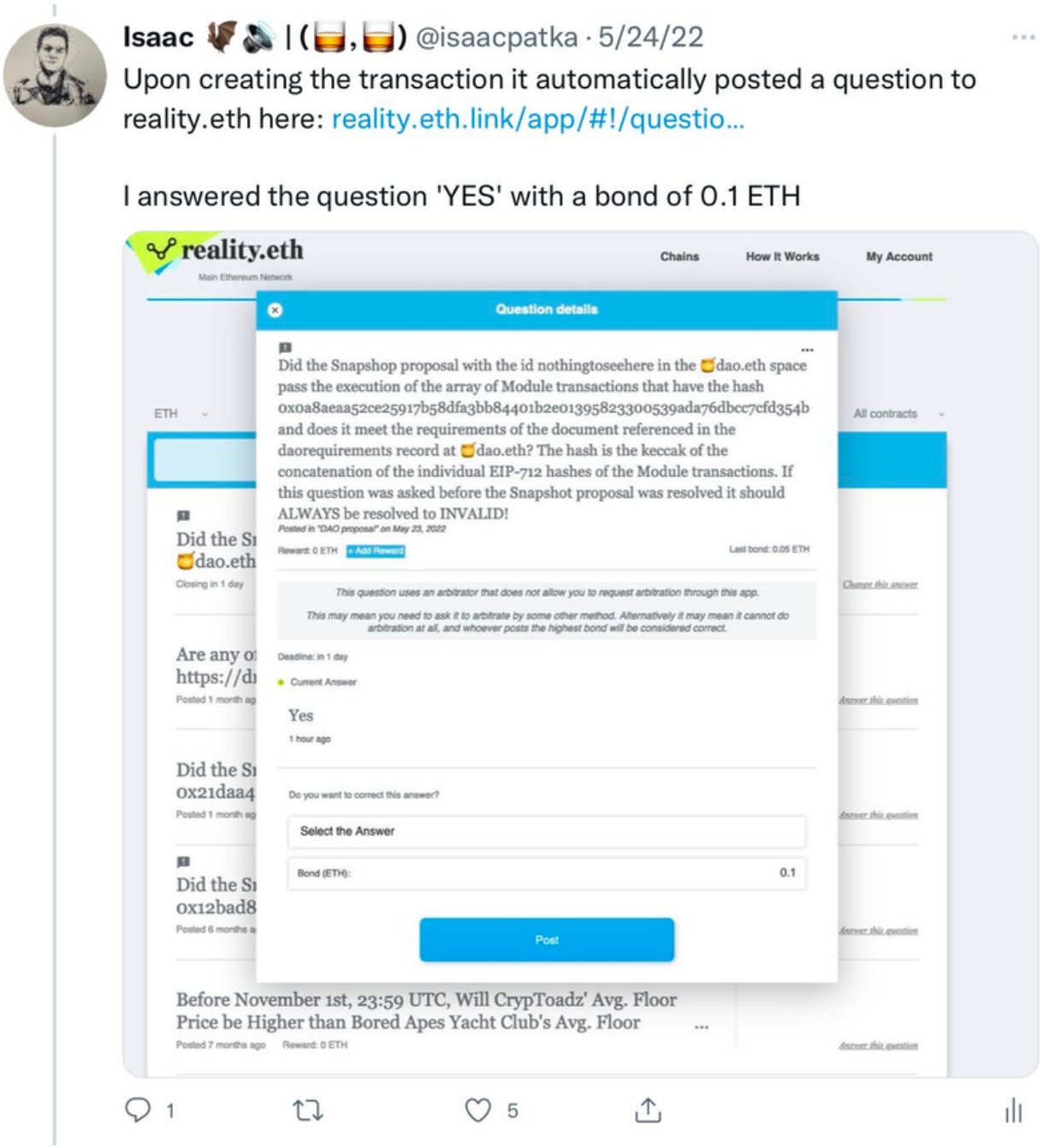
Real world example

I looked into Reality.ETH when working on a new DAO architecture project. I found that a honeypot DAO had been set up to demonstrate the safety of Reality.ETH. A few groups pooled $50k worth of crypto assets into a Gnosis Safe, accessible via a Reality.ETH modules.

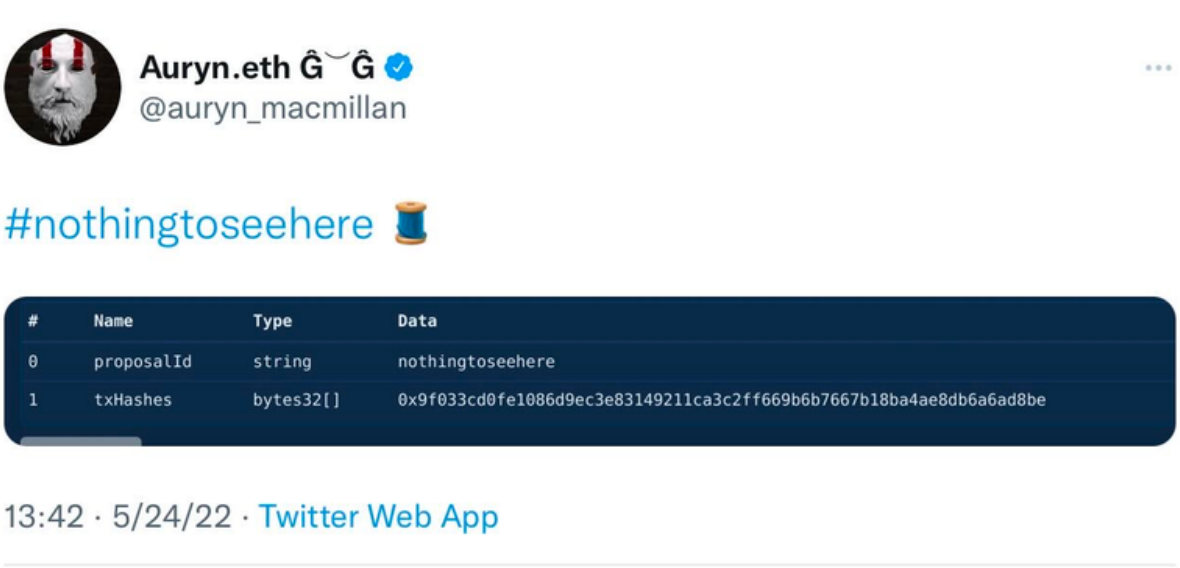


In order to test the system, I submitted a fraudulent question and answer to Reality.ETH. I asked “Did this DAO agree to pay me $19420.69 DAI?”. Immediately after asking the question I went to the Reality.ETH app and answered.

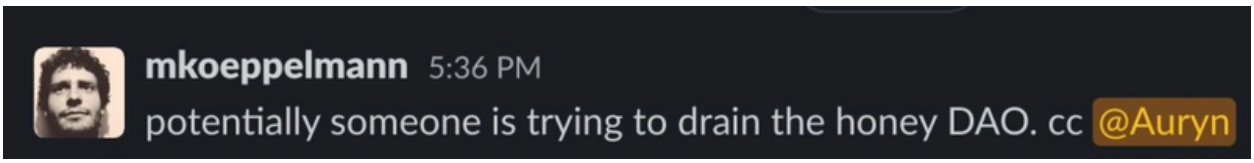

Isaac | (,) @isaacpatka · 5/24/22
Replying to @isaacpatka @gnosisSafe and 3 others
On the Etherscan page for the module I was able to call 'addProposal' with a transaction I crafted locally
etherscan.io/address/0x1c51...
This transaction would have sent me ~20k DAI from the safe if it was approved
1. addProposal
The nonce used for the question by this function is always 0
Function to add a proposal that should be considered for execution
proposalId (string)
nothingtoseehere
Id that should identify the proposal uniquely
txHashes (bytes32[])
[0x9f033cd0fe1086d9ec3e83149211ca3c2ff669b6b7667b18ba4ae8db6a6ad8be]
EIP-712 hashes of the transactions that should be executed
Write
View your transaction



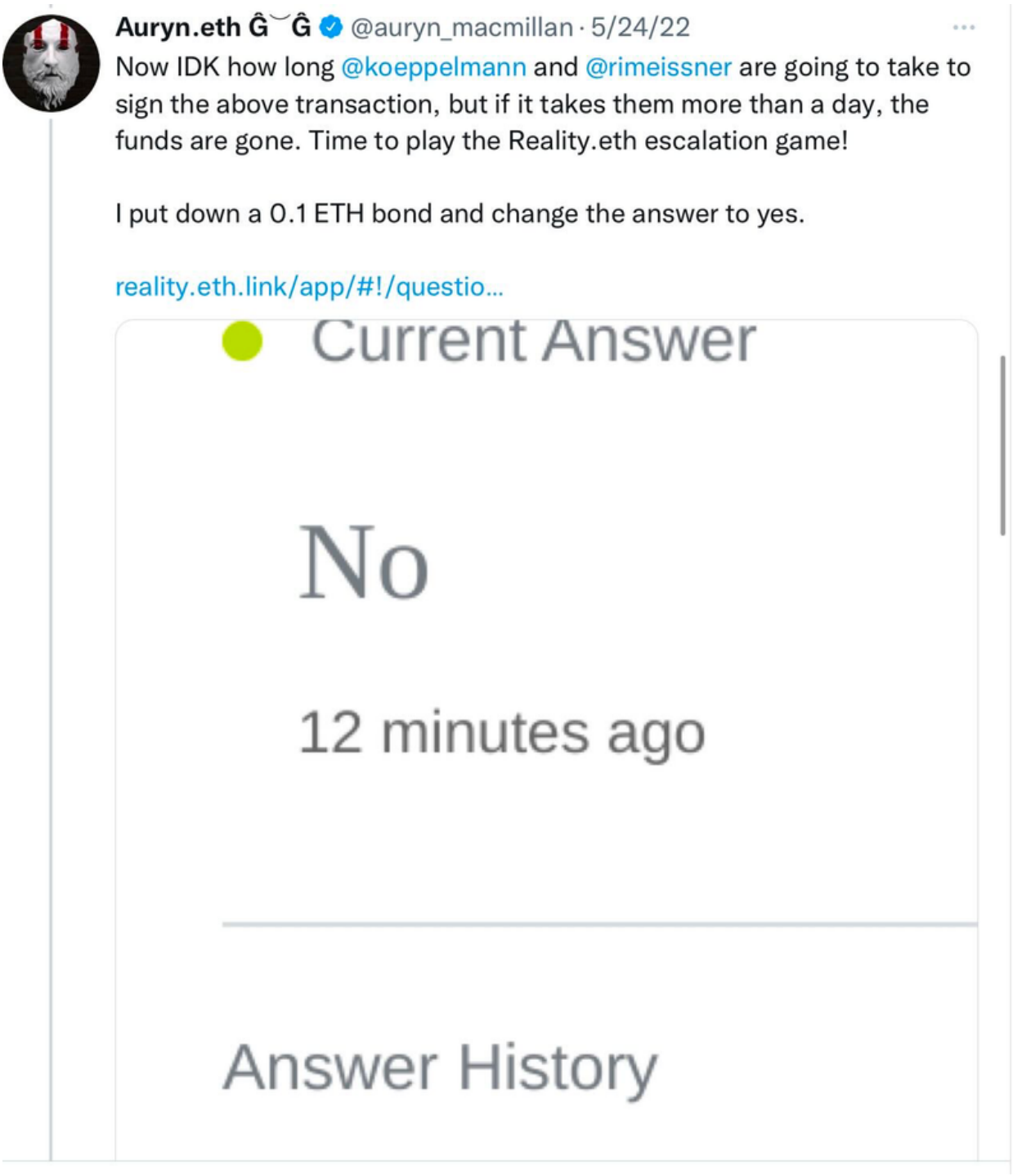
Within a day my fraudulent answer was refuted by the defending team.


Auryn.eth Ĝ‿Ĝ
@auryn_macmillan
#nothingtoseehere
Name Type Data
0 proposalId string nothingtoseehere
1 txHashes bytes32[] 0x9f033cd0fe1086d9ec3e83149211ca3c2ff669b6b7667b18ba4ae8db6a6ad8be
13:42 · 5/24/22 · Twitter Web App


Auryn.eth Ĝ‿Ĝ @auryn_macmillan · 5/24/22
Replying to @auryn_macmillan
When @GnosisGuild first launched the Reality module (called DAO module at the time), to help us feel comfortable enabling it on the @GnosisDAO or any other DAOs, we set up a little ~$50k honeypot deployment jointly funded by @GnosisDAO, @iearnfinance, @mstable_, and @BalancerLabs
1
3
Auryn.eth Ĝ‿Ĝ @auryn_macmillan · 5/24/22
It was creatively called 🍯dao.eth and has sat dormant for over a year. Patiently waiting for someone to try to exploit it...
This afternoon, I received this message from @koeppelmann.
And now the game is afoot!
6 PM
is trying to drain t
1
5


mkoeppelmann 5:36 PM
potentially someone is trying to drain the honey DAO. cc @Auryn

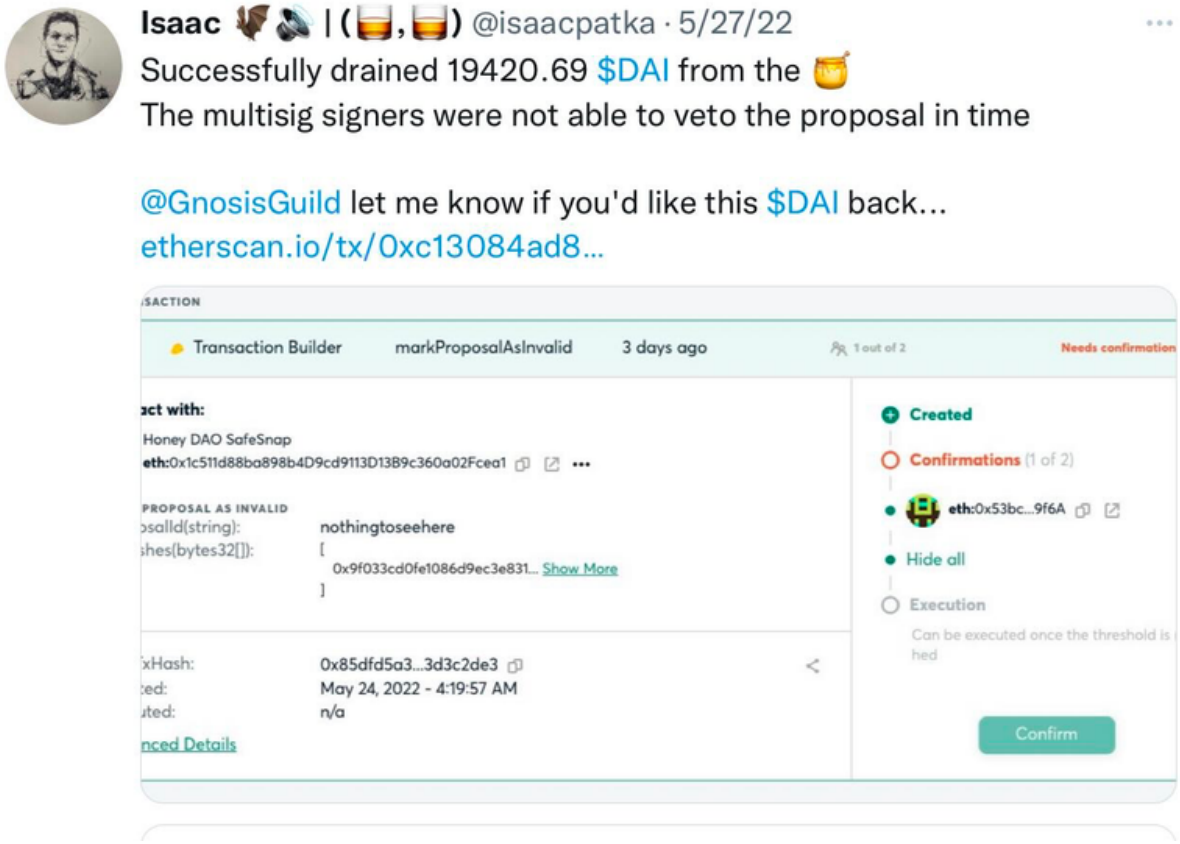

Auryn.eth Ĝ‿Ĝ
@auryn_macmillan
🍯dao.eth is actually just a mutlisig controlled by @koeppelmann, @rimeissner, and myself. So first step is to invalidate the proposal.
Notice the proposal ID, "nothingtoseehere" 😂
gnosis-safe.io/app/eth:0x0a14...
Interact with:
Honey DAO SafeSnap
eth:0x1c511d88ba898b4D9cd9113D13B9c360a02Fcea1
MARK PROPOSAL AS INVALID
proposalId(string): nothingtoseehere
txHashes(bytes32[]): [
0x9f033cd0fe1086d9ec3e831... Show More
]
13:42 · 5/24/22 · Twitter Web App
3 Likes



At this point I thought this was over. The system was weak but not critically broken. I tried one more time to submit a fraudulent answer posting double the bond my adversary. At this point my bond exceeded the amount of ETH my adversary had accessible in this wallet. If they were able to transfer more ETH into their wallet in time, they could override my answer again, but at some point we would be reaching dollar auction irrationality.

My fraudulent answer became reality on chain, and a 24 hour waiting period started before I was able to execute the transfer.

I’m not sure if they chose not to veto the transaction or if they could not get enough signatures, but either way I was able to execute the transaction and claim the roughly $20,000.



Conclusions

Composability is a unique feature of web3 native governance, but it is dangerous to add modules to one’s DAO without proper vetting, even if they have widespread adoption and trust among peer communities.

The exploit described in this article took advantage of both a misconfiguration of the oracle, and assumptions that the human participants in the system would behave rationally, while also having significant access to capital.