# Gaming the DAO

Emin Gün Sirer

Department of Computer Science
Cornell University

# DACs

- Decentralized Autonomous Corporations/Orgs are incredibly powerful and promising
- A computer program, with its own code and state, that can programmatically manage money flows
- The entire behavior of the program is pre-ordained
- Brand new era, with brand new functionality

# DAO Promise

- Automate and eliminate the middlemen
- Achieve far higher efficiencies
  - A hedge fund with 0% overhead?
- Self-policing and/or self-arbitrating
  - Can't eliminate the legal system, but can handle simple cases
- Bring complete transparency to the operation of a company or trust
  - Insurance
  - Finance
- Killer apps are yet to come…

# DAO Unknowns

## Is it actually possible to build secure, functional smart-contracts?*

* what about the fine print you see on regular contracts?

What's in the fine print?

How to form the contract covenant
The spirit of the agreement
How to resolve disputes
How to modify the contract
How to terminate

**The DAO, as we will see, messed up almost all of these**

# Enter The DAO

- Usurped the phrase “The DAO” for a specific investment fund
- Part kickstarter, part Andreesen-Horowitz
  - Built by Slock.It, a company originally intended to kickstart an IoT bike lock, but built a kickstarter instead
- How it is supposed to work
  - We all buy into The DAO with ether
  - The DAO amasses a fund
  - Contractors come before The DAO with proposals
  - We all vote on the proposals
  - If we achieve a quorum, and there is support, proposals get funded
  - Proposals then return rewards, distributed back out

# The DAO Complications

- Buying in
- Voting
- Exiting
- Modifying the Contract
- Payouts

# The DAO Buy-In

- 27-day creation phase
- Buy in with ether
  - 1.00 ether for 100 **DAO Tokens** for 14 days
  - +0.05 ether every day for 10 days
  - 1.50 ether for the last 3 days
- Additional gains accumulate in “extraBalance”
- Why is there a rising scale?
- Do “viral features” have any place in sound investments?

# The DAO Proposals

- Anyone can submit a **proposal**
- **Curators** pick proposals
  - Requires a 5 out of 11 signature
  - 11 members of the Ethereum community, unrelated to SlockIt
- The curators' job description is unclear
  - Is it to just check identity?
  - Is it to "protect the DAO"?
- The curators are not paid, but they are under substantial legal risk

# The Voting

- Any DAO token holder can vote on a proposal
- A proposal is funded if
  - There is a quorum (sufficient votes)
  - The majority of the quorum is in favor (voted YES)
- Required quorum sizes vary by size of contract
  - Largest required quorum is 53%
- Votes are weighted by a voter's holdings
- But a voter commits The DAO funds (i.e other people's money) to proposals
- Someone who voted cannot exit The DAO

# The Exit

- Cannot just take money out of The DAO
  - Why? Because of viral/social reasons

- To exit, you need to follow a **62-step process**:
  - Initiate a proposal to make yourself a curator
  - Anyone can vote YES or NO on this proposal
  - It will likely fail
  - You can call **splitDAO** on a failed proposal
  - A new child-DAO will be created where you are the curator
  - You can now propose to withdraw funds, approve it as curator, vote on it, and then take the ether back out

- Takes 27+7 days

# Upgrades and Rewards

- There is **no provision to modify The DAO** in place
  - No kill switch
  - No security upgrades
  - Cannot preserve the full state and change code
- The extraBalances can only be spent after The DAO has spent an equivalent amount on proposals
- Unclear about the intended behavior with regard to **rewards**
  - Inherited into childDAO's, but not into grandchildren

# The DAO Token Markets

- DAO tokens can be bought and sold on open markets
- Their price will reflect the expected value of future ether flows
- Until The DAO funds a proposal, 1 token = 0.01 eth
- But in USD terms, the price will fluctuate
- The price difference will reflect the uncertainty in the value of 1 eth, 34 days from now
  - E.g. 1 eth = $15
  - But 1 dao = $13
- This is a normal consequence of decisions in DAO design

# Taking Stock

- Why was The DAO designed the way it was?
  - To avoid legal meddling?
  - To help fund illegal operations?
  - To create Ponzis?
- “Sunny-day thinking”
- Aspirational system design
- Does The DAO idea even make sense?

# The Questions

- Are the crowds even able to pick winning strategies?
  - Do fund managers really bring 0 value to the world?
- Will we ever reach the quorums required?
  - Most token holders are passive
  - The risks of “going with the crowd” without voting
- Are the mechanisms in The DAO suited for the tasks that need to be carried out?

# The Questions

- Are the crowds even able to pick winning strategies?
  - Do fund managers really bring 0 value to the world?
- Will we ever reach the quorums required?
  - Most token holders are passive
  - The risks of “going with the crowd” without voting
- Are the mechanisms in The DAO suited for the tasks that need to be carried out?
  - NO!

# The Call for a Moratorium

- My colleagues and I were alarmed that The DAO managed to collect 11M eth, $220M USD
- The internal mechanisms were broken
- We rushed a manuscript that detailed the failures, called for a moratorium
- The DAO community was convinced and wanted to upgrade The DAO

# The Hack

- While we were in a holding pattern, someone emptied out a substantial fraction of The DAO
- The hacker took $50+M worth of ether into a child-DAO called the Dark-DAO
- Hacker took advantage of multiple attack vectors
  - A reentrancy bug in the DAO code
  - Additional tricks to avoid getting his balance reset
  - He also voted YES on every other split proposal, to reserve the right to pursue everyone who wanted to split
- Hide your kids, hide your pets, there is no safe place

# The Hack Technicalities

```
// Assign reward rights to new DAO
uint rewardTokenToBeMoved =
    (balances[msg.sender] * p.splitData[0].rewardToken) /
    p.splitData[0].totalSupply;

uint paidOutToBeMoved = DAOpaidOut[address(this)] * rewardTokenToBeMoved /
    rewardToken[address(this)];

rewardToken[address(p.splitData[0].newDAO)] += rewardTokenToBeMoved;
if (rewardToken[address(this)] < rewardTokenToBeMoved)
    throw;
rewardToken[address(this)] -= rewardTokenToBeMoved;

DAOpaidOut[address(p.splitData[0].newDAO)] += paidOutToBeMoved;
if (DAOpaidOut[address(this)] < paidOutToBeMoved)
    throw;
DAOpaidOut[address(this)] -= paidOutToBeMoved;

// Burn DAO Tokens
Transfer(msg.sender, 0, balances[msg.sender]);
withdrawRewardFor(msg.sender); // be nice, and get his rewards
totalSupply -= balances[msg.sender];
balances[msg.sender] = 0;
paidOut[msg.sender] = 0;
return true;
```

# What If The DAO Had Not Been Hacked

- It still would have been hacked
- It was and is deeply broken
- The design of voting mechanisms that capture the will of the crowds is a difficult nuanced task
- Everybody on the Internet is an expert at three things:
  - Economics
  - Game theory
  - Distributed Systems
- The DAO team, and others like it, full of hubris and the Dunning-Kruger effect, are easy targets

# Guiding Principle

- DAO-1.0 is irredeemably broken, but let's examine how one might build DAO-2.0 in light of what we have learned
- The DAO voting mechanisms have to be **truthful** and **strategy-proof**
  - Truthful: token holders vote their true opinion
  - Strategy-proof: token holders fare best by voting their true opinion
- The current mechanisms are broken in multiple ways

# Affirmative Bias

- Every voter has a unique valuation for every proposal
  - “Prop #37 will bring in 3% yearly over 3 years”
  - “Prop #37 will be a net loss, that team can’t pull it off”
  - “Prop #37 will take us to the moon!”

- Ideally, you want everyone to vote their conscience
  - Positive Expected Value: +EV
  - Negative Expected Value: -EV

- +EV folks are incentivized to vote early

- Not so for -EV!!!
  - Negative votes lock people in

- Early votes will be positive, feedback loops work against -EV folks

# Stalking

- A stalker can vote YES on a split proposal and follow a splitter into the child-DAO
- Stalker is not going to be the curator, but he can be the dominant (53%) shareholder in the child
- Stalker can keep the splitter from taking out his funds
- Stalker can then blackmail the splitter
- If the splitter splits again, he loses his rewards from the original DAO
- SlockIt claimed that the splitter could counterattack, but do you want to play corewars?

# Ambush

- A -EV voter has a disincentive to vote, especially if his vote is not needed
- So a big bloc of YES votes can come in at the last possible minute to pass a proposal that initially looked unpassable
- This commits other people's funds to a proposal, even though large fraction is against that proposal
- Possible remedy: add time to the clock when the vote outcome changes

# Token Raid

- An attacker can move the price of DAO tokens by
  - Incentivize people not to split but to sell their tokens
  - Keep the public from snapping up tokens
- She can do this by
  - Creating social media panic, via stalker attack
  - Passing a -EV proposal, via ambush attack
- The price of tokens will drop, she can short on the way down, and snap up when the attack is over
- This is a legitimate manipulation strategy, often seen with penny stocks, except the mechanisms make it easy

# extraBalance Raid

- Attacker forces people to split from The DAO, which leaves behind the extraBalance amount
- Currently at 275,000 ether
- DAO tokens should trade at 1.02
- If the attacker scares away 95% of investors, DAO will trade at 2.00

# Majority Takeover

- SlockIt identified and worried about a majority takeover
- A voting bloc of 53+% can fund 100% to a 1 proposal
- Curators are expected to guard against this
  - This scenario is specifically cited
- But a voting bloc of 53+% can fund 10 proposals of 10%
- No principled way to even define the attack, let alone defend against it
  - DAO defenseless against Soros-style attacks

# Reward Dilution

- The DAO issues reward tokens as proposals pay back into the DAO
- Akin to dividends
- But the reward token math does not follow any accounting principle
- In particular, reward tokens can be diluted even after someone has split off from the DAO

# Risk-Free Voting

- One of the many “race conditions”
- Investor votes YES on a proposal, committing funds
- Then invokes “unblockMe” before the proposal is executed, and splits off
- This allows her to commit the DAO to a proposal without committing her own funds
- An attack amplification vector

# Concurrent Proposal Trap

- Voting on any proposal commits the voter until the end of the voting period
- Attacker poses a proposal
  - We have seen “do you believe in God?” for 0 ether
- Everyone who votes is banned from splitting until the end of the voting period
- Attack amplification vector: push an incendiary proposal with a long voting period, then launch short-fuse attack

# Independence Assumption

- All of the discussion until now assumes that all proposals are independent
- Yet in real life, proposals are linked
  - Funding a cluster of proposals might yield much higher returns than funding them individually
- Not an attack, but undesirable
- This can yield strategic behavior (i.e. people voting down worthy proposals) even when everyone means well

# What Have We Learned

- The DAO is a fantastic experiment
- The experiment has been a huge success
- Enormous demand for smart contracts
- The Ethereum core has some (well-contained) issues that need to be fixed
  - The design of a secure smart-contract language is very different from the design of a web-programming language
- The DAO is a hot mess

# Methodological Issues

- Why was The DAO designed the way it was?
  - To avoid legal meddling?
  - To help fund illegal operations?
  - To create Ponzis?
- Carefully thought-out viral features
- Common behaviors were purposefully made difficult
- “Sunny-day thinking,” aspirational ideas about best case behaviors
- Irresponsible design, no safety mechanisms
- Flawed methodology

# Takeaways

- Can we build a $1.2B ecosystem, while spending $0 on basic research and science of smart contracts?
- How do we build and vet trustworthy smart contracts?
- IC3, Initiative on Cryptocurrencies and Smart Contracts, http://initc3.org