The DAO Attack

or, How To Steal $60 Million With Smart Contracts

By Charles Lin for CS 261
Tuesday, September 12, 2018
Today, we are covering...

- Decentralized Autonomous Organizations (DAOs)
- The DAO Attack: story and technical details
- Aftermath and implications
What is a Decentralized Autonomous Organization?
What is a Company?
What is a Company?

- A legal entity that people can purchase shares of
- People can use shares to vote on how to spend company assets
- Behaviour enforced by people
- Reneging punished by law
1. Introduction

Corporate entities of all kinds are governed by rules that describe permitted and proscribed conduct. These rules may exist as private contracts (like bylaws or shareholder agreements) between corporate owners. They may also be imposed by law in addition to or in the absence of a written agreement between participants.

Historically, corporations have only been able to act through people (or through corporate entities that were themselves ultimately controlled by people). This presents two simple and fundamental problems. Whatever a private contract or public law require: (1) people do not always follow the rules and (2) people do not always agree what the rules actually require. Collaboration without a corporate form does not solve these problems, necessarily, and it may introduce others. In the absence of a corporate form, an explicit written agreement is substituted for unclear informal “understandings” and the legal protections provided by a corporate form will not be available.

Rule-breaking within an organization not always obvious, and motives may not matter to stakeholders once their money has been lost. While bad behavior may make a corporation or its management civilly or criminally liable, punishment can come as little comfort to an investor who has already lost their money.

- People don’t always agree on the rules
- People don’t always follow the rules
- Legal punishment is a deterrent, not absolute enforcement, and does not repair damage
What is a Company?

- A legal entity that people can purchase shares of
- People can use shares to vote on how to spend company assets
- Behaviour enforced by people
- Reneging punished by law
What is a Company?

- A **legal entity** that people can purchase shares of
- People can use shares to vote on how to spend company assets
- Behaviour **enforced by people**
- Reneging **punished by law**
What is a Decentralized Autonomous Organization?

- A **contract** that people can purchase shares of
- People can use shares to vote on how to spend **contract’s ether balance**
- Behaviour **enforced by code**
- Reneging **impossible!**
“The” DAO
slock.it and “The DAO”

- April 2016
- German startup called slock.it - smart locks connected to blockchain
- Started a venture capital DAO to get the community to fund their startup and other startups of their choice
- Called it “The DAO”
THE DAO IS REVOLUTIONARY.
The time has come to breathe life into The DAO
A One Time Only Event

107.17 M
DAO TOKENS CREATED

1.07 M
TOTAL ETH

9.23 M
USD EQUIVALENT

1.00
CURRENT RATE
ETH / 100 DAO TOKENS

12 days
UNTIL NEXT PRICE CHANGE

26 days
LEFT
ENDS 28 MAY 09:00 GMT

To obtain DAO tokens, follow the wizard below or send ETH from your Ethereum Wallet (NOT an exchange) to The DAO’s address below. Note that sending ETH to the DAO’s address signifies your acceptance of the Terms.

0xbb9bc244d798123fde783fcc1c72d3bb8c189413
DO NOT SEND ETH TO THIS ADDRESS DIRECTLY FROM AN EXCHANGE. USE THE ETHEREUM WALLET OR FOLLOW THE WIZARD BELOW

START WIZARD
The DAO Code (simplified)

```solidity
contract SimpleDAO {
    mapping (address => uint) public credit;
    function donate(address to){credit[to] += msg.value;}
    function queryCredit(address to) returns (uint){
        return credit[to];
    }
    function withdraw(uint amount) {
        if (credit[msg.sender]>= amount) {
            msg.sender.call.value(amount)();
            credit[msg.sender] -= amount;
        }
    }
}
```

from Atezi et al.
This looked secure.
Deja Vu DAO Smart Contracts Audit Results

We’re pleased to announce that one of the world’s leading security audit companies, Deja Vu Security, has performed a security review of the generic DAO framework smart contracts.

Obvious puns aside, the name Deja Vu Security should be familiar, as it is their experts that completed the Ethereum codebase audit back in July 2015. To say the quality of their work is top notch is an understatement.
This looked secure.
This looked secure.

This was not secure.
How was it insecure?

- Nuances of Solidity behaviour
- Transactions of ether can only be done by tacking on to a function call
- If no function specified, recipient’s “default” function is ran
- Recipient can maliciously craft their default function!
“Recursive call vulnerability”

```solidity
contract Mallory {
    SimpleDAO public dao = SimpleDAO(0x354...);
    address owner;
    function Mallory(){owner = msg.sender; }
    function() { dao.withdraw(dao.queryCredit(this)); }
    function getJackpot(){ owner.send(this.balance); }
}
```

from Atezi et al.
PLEASE
CONTINUE...
More Ethereum Attacks: Race-To-Empty is the Real Deal
09 JUNE 2016 on ethereum, smart contracts, security, solidity

Chriseth at github casually pointed out a terrible, terrible attack on wallet contracts that I had not considered. If there were a responsible disclosure avenue for ethereum contract developers, I would use it, but there doesn't seem to be. Not only that, this code has been out and published on github for long enough that I wanted to get the news out there quickly.

In Brief: Your smart contract is probably vulnerable to being emptied if you keep track of any sort of user balances and were not very, very careful.

https://vessenes.com/more-ethereum-attacks-race-to-empty-is-the-real-deal/
No DAO funds at risk following the Ethereum smart contract ‘recursive call’ bug discovery

Our team is blessed to have Dr. Christian Reitwießner, Father of Solidity, as its Advisor. During the early development of the DAO Framework 1.1 and thanks to his guidance we were made aware of a generic vulnerability common to all Ethereum smart contracts. We promptly circumvented this so-called “recursive call vulnerability” or “race to empty” from the DAO Framework 1.1 as can be seen on line 580:

https://blog.slock.it/no-dao-funds-at-risk-following-the-ethereum-smart-contract-recursive-call-bug-discovery-29f482d348b
CRITICAL UPDATE Re: DAO Vulnerability

Posted by Vitalik Buterin on June 17, 2016

An attack has been found and exploited in the DAO, and the attacker is currently in the process of draining the ether contained in the DAO into a child DAO. The attack is a recursive calling vulnerability, where an attacker called the “split” function, and then calls the split function recursively inside of the split, thereby collecting ether many times over in a single transaction.

https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/
How bad was it?

- The DAO was virally popular
- It was holding 15% of all ether at the time ($150 mm/ $1bn)
- $60 million was stolen by first hacker
- “white hat hackers" took the rest

Discussion: What do you think the right thing to do here is? Is there even a right thing to do?
Arguments for intervention

- Trust and stability in Ethereum
- It's experimental technology and it won't take off the ground if adopters don't feel safe
- Legal worries - class action lawsuit?
- Obviously, Ethereum foundation was very much on this side
- (Should note that foundation staff owned many DAO shares)
Arguments against intervention

- Goes against the whole idea of Ethereum
Build unstoppable applications

Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.
"The involvement of the ethereum foundation in the DAO has been and is a mistake. As I see it, ethereum is supposed to be the foundational infrastructure upon which a flurry of projects and experiments are supposed to blossom, and in order for them to blossom, they need a foundation that is strong, and that has integrity in the face of challenges. The hard fork proposal is a compromise that ruins that integrity and signals that projects like the DAO can influence the underlying foundation to their own advantage. To me, that is totally unacceptable and is a departure from the principles that drew me to ethereum."
"I made a bad contract in the first days ETH was online and lost 2K ETH with it, can I also get it back? Thanks!"
"Ethereum worked exactly as intended. I don't believe software should be updated when it works exactly as intended. You assume the risks of your investment. If you don't understand your investment, you assume unknown risk. Anything else is a bailout by a central authority, i.e. the antithesis of the crypto world. In a related way, this is why Lehman Brothers was allowed to fail – because the deal is the deal, and if you bend the rules for a particular player, all other players will want special treatment, too."
BEGIN SIGNED MESSAGE

To the DAO and the Ethereum community,

I have carefully examined the code of The DAO and decided to participate after finding the feature where splitting is rewarded with additional ether. I have made use of this feature and have rightfully claimed 3,641,694 ether, and would like to thank the DAO for this reward. It is my understanding that the DAO code contains this feature to promote decentralization and encourage the creation of "child DAOs".

I am disappointed by those who are characterizing the use of this intentional feature as "theft".

via Pastebin
Ethereum Foundation decided to intervene

- Discussion: How?
- Soft fork: Client software update to blacklist the account of the thief
- Had security vulnerabilities and was dropped as an option
- **Hard fork:** Write a hardcoded block closing down the DAO, restoring all funds to original owners, and forcing everyone to accept that block
• New client shipped with user option to accept hardcoded block at 1920000
- New client shipped with user option to accept hardcoded block at 1920000
- 85% of clients accepted
• New client shipped with user option to accept hardcoded block at 1920000
• 85% of clients accepted
• 15% of clients rejected and continued mining on the old chain
The Aftermath
Let it be known to the entire world that on July 20th, 2016, at block 1,920,000, we as a community of sovereign individuals stood united by a common vision to continue the original Ethereum blockchain that is truly free from censorship, fraud or third party interference. In realizing that the blockchain represents absolute truth, we stand by it, supporting its immutability and its future. We do not make this declaration lightly, nor without forethought to the consequences of our actions.
- **Recommendations for Smart Contract Security in Solidity**
  - External Calls
    - Avoid external calls when possible
    - Use `send()`, avoid `call.value()`
    - Handle errors in external calls
    - Don’t make control flow assumptions after external calls
    - Favor `pull` over `push` for external calls
    - Mark untrusted contracts
  - Beware rounding with integer division
  - Remember that on-chain data is public
  - In 2-party or N-party contracts, beware of the possibility that some participants may "drop offline" and not return
  - Keep fallback functions simple
  - Explicitly mark visibility in functions and state variables
  - Beware division by zero
  - Differentiate functions and events

- **Known Attacks**
  - Call Depth Attack ( Deprecated )
  - Race Conditions*
    - Reentrancy
    - Cross-function Race Conditions
    - Pitfalls in Race Condition Solutions
  - DoS with (Unexpected) Throw
  - DoS with Block Gas Limit
  - Timestamp Dependence
  - Transaction-Ordering Dependence ( TOD )

[Link to Safety page on Ethereum Wiki](https://github.com/ethereum/wiki/wiki/Safety)
Conclusion
We’ve gone over

- the concept of DAOs,
- the details of the DAO attack,
- the resulting ideological rifts.

Insights? Paraphrasing slock.it:

- Smart contract security, like all types of security, is pretty hard to get right! Need to actively work on better tooling, practices, lower complexity, etc.
- No matter what: fast, controversial decisions is bad for decentralized trust. Decentralized governance of Ethereum is an open problem!
Thank you! :)