Case 3: The DAO Hack

Module 3: Can We Build Trust Without Intermediaries?

Background

A DAO, or Decentralised Autonomous Organisation, is an entity governed entirely by code running on a blockchain. Instead of a board of directors, bylaws, and corporate officers, a DAO uses smart contracts (self-executing programs stored on a blockchain, as covered in M3L3) to encode its rules. Token holders vote on proposals, and the smart contract executes the outcome automatically. No lawyers, no boardrooms, no trust in individuals required – only trust in the code.

"The DAO" (capitalised to distinguish it from the general concept) was the first large-scale attempt to build such an organisation. Created in April 2016 on the Ethereum blockchain, it was designed as a decentralised venture capital fund. Anyone could buy DAO tokens with Ether (ETH, Ethereum's native cryptocurrency), and token holders would collectively vote on which projects to fund. The DAO had no management team, no physical office, and no legal jurisdiction. Its creators at a German company called Slock.it described it as an experiment in trustless governance.

Between April 30 and May 28, 2016, The DAO raised approximately USD 150 million in ETH from roughly 11,000 investors, making it the largest crowdfunding campaign in history at that time. The enthusiasm was enormous. The DAO held roughly 14% of all Ether in circulation.

What Happened

On June 17, 2016, an unknown attacker began draining funds from The DAO. The attack exploited a vulnerability known as a reentrancy bug in The DAO's smart contract code. Here is how it worked in simplified terms:

The DAO's code allowed investors to withdraw their funds through a "split" function. This function was supposed to (1) send the requested ETH to the caller and then (2) update the caller's balance to reflect the withdrawal. The vulnerability was in the ordering: the code sent the ETH *before* updating the balance. The attacker wrote a malicious contract that, upon receiving ETH in step 1, immediately called the split function again – before step 2 could execute. This created a loop: withdraw, re-enter, withdraw, re-enter. Each cycle drained funds without the balance ever being decremented.

The attacker siphoned approximately USD 50 million (3.6 million ETH) into a "child DAO" – a separate contract. Due to The DAO's own rules, the funds were locked in this child DAO for 28 days before they could be moved further, creating a window for the community to respond.

The Ethereum community faced a dilemma with no good options. The bug was not in Ethereum itself but in The DAO's application code. Three responses were debated:

Option 1: Do nothing. The code executed as written. "Code is law" – the foundational principle of smart contracts – meant the attacker had simply found a legitimate (if unintended) use of the contract. Intervening would undermine the entire premise of trustless systems.

Option 2: Soft fork. Freeze the attacker's funds by blacklisting the child DAO address. This would prevent the attacker from moving the ETH but would not return funds to investors.

Option 3: Hard fork. Rewrite Ethereum's transaction history to move all DAO funds (including the stolen portion) to a new recovery contract, allowing investors to withdraw their original stakes. This would reverse the theft but would also reverse the blockchain's immutability – the property that no transaction, once recorded, can be altered.

On July 20, 2016, the Ethereum network executed a hard fork. Approximately 85% of miners supported the fork. The stolen funds were returned. But a minority of the community refused to accept the rewrite. They continued running the original, unforked chain, which became known as Ethereum Classic (ETC). As of 2025, both chains still operate independently.

The Analysis

The DAO hack exposed a fundamental tension at the core of M3's question: if we replace trusted intermediaries with code, the code itself becomes the single point of failure. Traditional venture capital funds have lawyers who review contracts, compliance officers who flag risks, and regulators who enforce rules. The DAO replaced all of these with a smart contract that contained a bug no one caught before deployment.

The reentrancy vulnerability was not novel. Security researchers had warned about it in public forums before the attack. A paper by researchers at the University of Maryland had flagged reentrancy as a general risk in Solidity (Ethereum's programming language). But The DAO launched without a formal security audit by an independent third party – a practice that has since become standard for major smart contract deployments.

The hard fork decision revealed that "trustless" systems still depend on social consensus. When the code produced an outcome the community found unacceptable, the community overrode the code. This re-introduced human judgment – exactly the intermediary function that blockchains were designed to eliminate. The fork worked because Ethereum was still young and its community small enough to coordinate. Whether such coordination would be possible on a larger, more decentralised network is an open question.

The DAO hack accelerated the development of smart contract security practices: formal verification (mathematically proving code correctness), security audits, bug bounty programmes, and more conservative contract design patterns. It also informed regulatory thinking worldwide. The US Securities and Exchange Commission (SEC) issued a report in July 2017 concluding that DAO tokens were securities under US law, establishing a precedent that decentralised governance does not exempt a project from securities regulation.

Discussion Questions

1. Using M3L3 (Smart Contracts), explain what a reentrancy vulnerability is and why the ordering of "send funds" and "update balance" in The DAO's code created the exploit.
2. The Ethereum community chose a hard fork that reversed the theft but violated blockchain immutability. Evaluate whether this decision strengthened or weakened trust in Ethereum as a platform, using the trust framework from M3L1 (Cryptographic Foundations).
3. The SEC concluded in 2017 that DAO tokens were securities. How does this connect to the regulatory classification challenges discussed in M7L3 (Regulating the New), and what are the implications for current DeFi governance tokens?
4. If you were a smart contract developer launching a decentralised fund today, what technical and organisational safeguards would you implement that The DAO lacked?
5. Could a DAO hack of this scale happen again given current smart contract security practices, or has the ecosystem matured enough to prevent it?

Further Reading

- Meier, R. and Malone, T. (2017). "The DAO: A Case Study." MIT Sloan School of Management Working Paper.
- US Securities and Exchange Commission (2017). *Report of Investigation Pursuant to Section 21(a): The DAO*. Release No. 81207.
- Siegel, D. (2016). "Understanding The DAO Attack." *CoinDesk*, June 25, 2016.