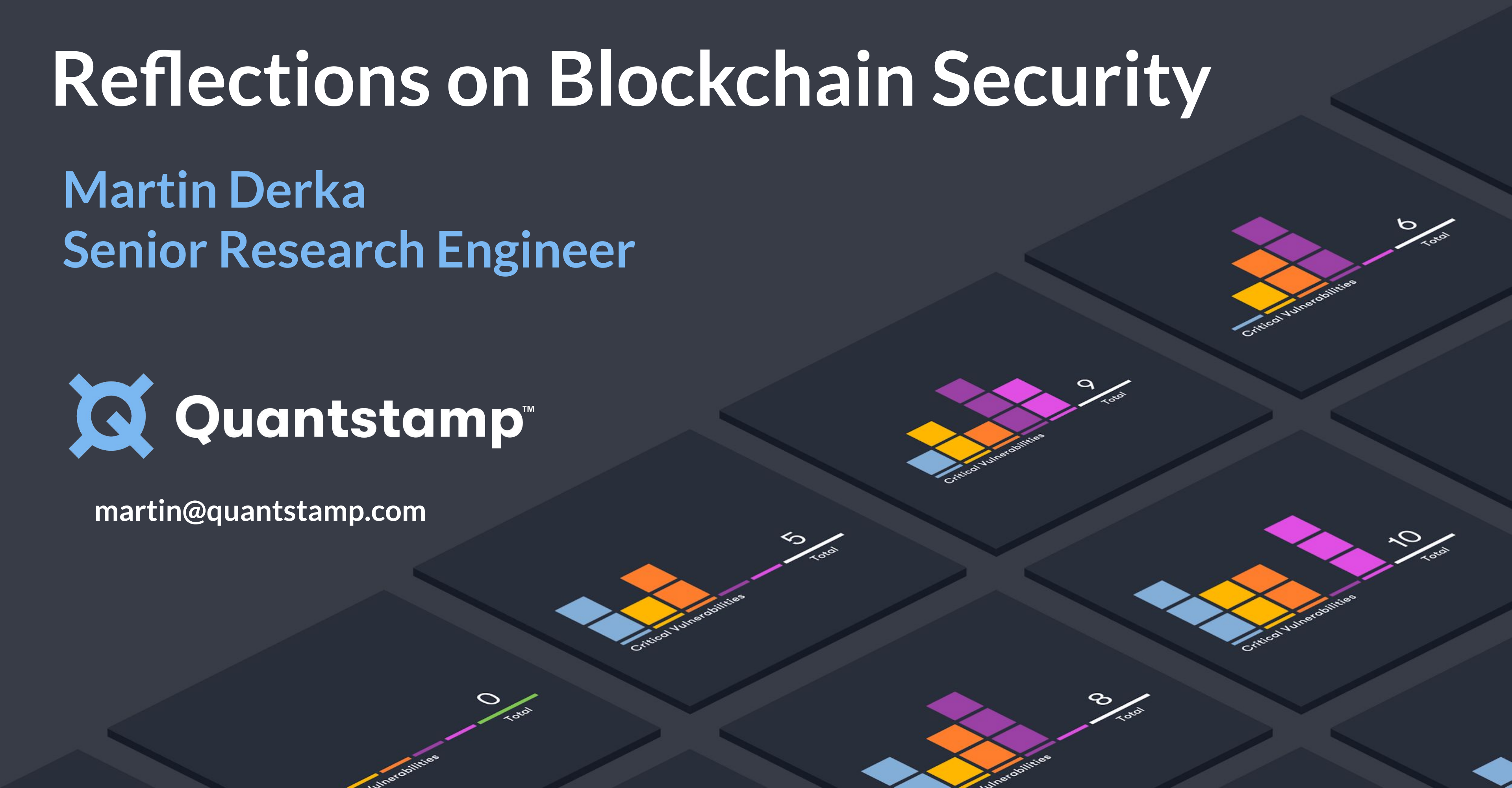
Reflections on Blockchain Security

Martin Derka
Senior Research Engineer

Quantstamp™

martin@quantstamp.com



Topic

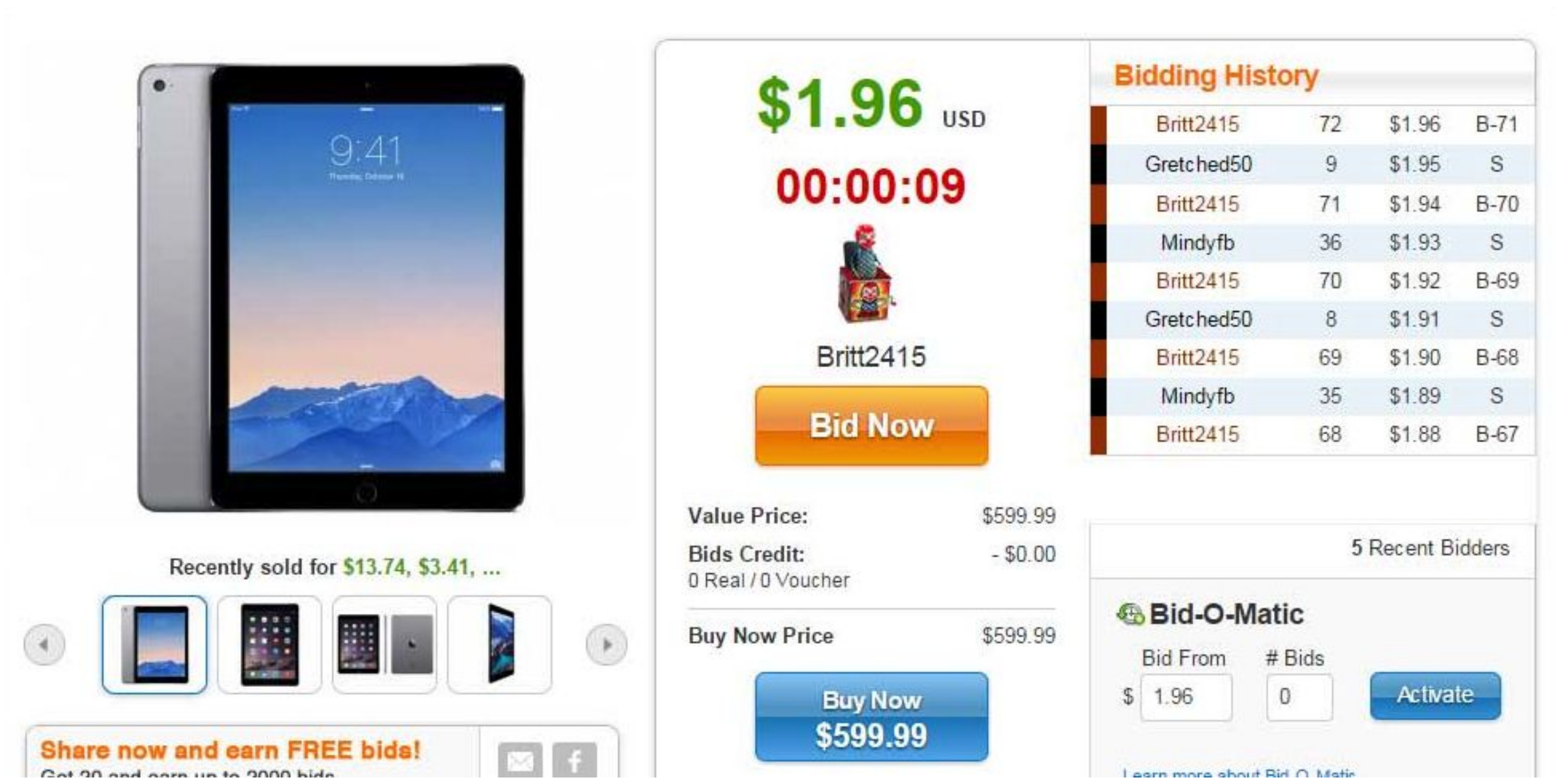
Mining Manipulation

Influencing the order or timeliness of transaction mining

Case Studies:

- Fomo3D (Network Jamming)
- FairWin (Front Running)

Penny Auctions



Recently sold for $13.74, $3.41, ...

Share now and earn FREE bids!

Get 20 and earn up to 2000 bids

$1.96 USD

00:00:09

Britt2415

Bid Now

Value Price:			$599.99
Bids Credit: 0 Real / 0 Voucher			- $0.00
Buy Now Price			$599.99

Buy Now $599.99

Bidding History

Britt2415	72	$1.96	B-71
Gretched50	9	$1.95	S
Britt2415	71	$1.94	B-70
Mindyfb	36	$1.93	S
Britt2415	70	$1.92	B-69
Gretched50	8	$1.91	S
Britt2415	69	$1.90	B-68
Mindyfb	35	$1.89	S
Britt2415	68	$1.88	B-67

5 Recent Bidders

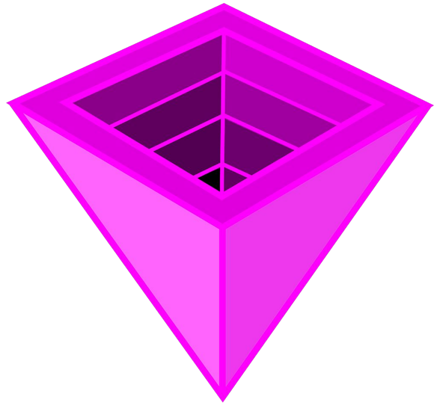
Bid-O-Matic

Bid From	# Bids	
$ 1.96	0	Activate

Learn more about Bid-O-Matic

Fomo 3D

1. The last investor takes the jackpot
2. Investments extend the timer



Fomo 3D:
A ponzi scheme, smart contract resembling a penny auction
https://etherscan.io/address/0xa62142888aba8370742be823c1782d17a0389da1

Fomo 3D

- Deployed on July 6, 2018
- Over 17,000 ETH by July 21, 2018 ($5.1 million @ $300 per ETH)

One possible outcome [...] is the game will attract a huge amount Ethers, and can do it again and again. [...] Ultimately, it will drain all liquidity of Ether and let the winner takes all. Then Ethererum's network value will be largely destroyed. [...]

toliuyi, Ethresear.ch forum

Fomo 3D

- Deployed on July 6, 2018
- Over 17,000 ETH by July 21, 2018 ($5.1 million @ $300 per ETH)

Miner pool collusion much more likely.

Fomo 3D

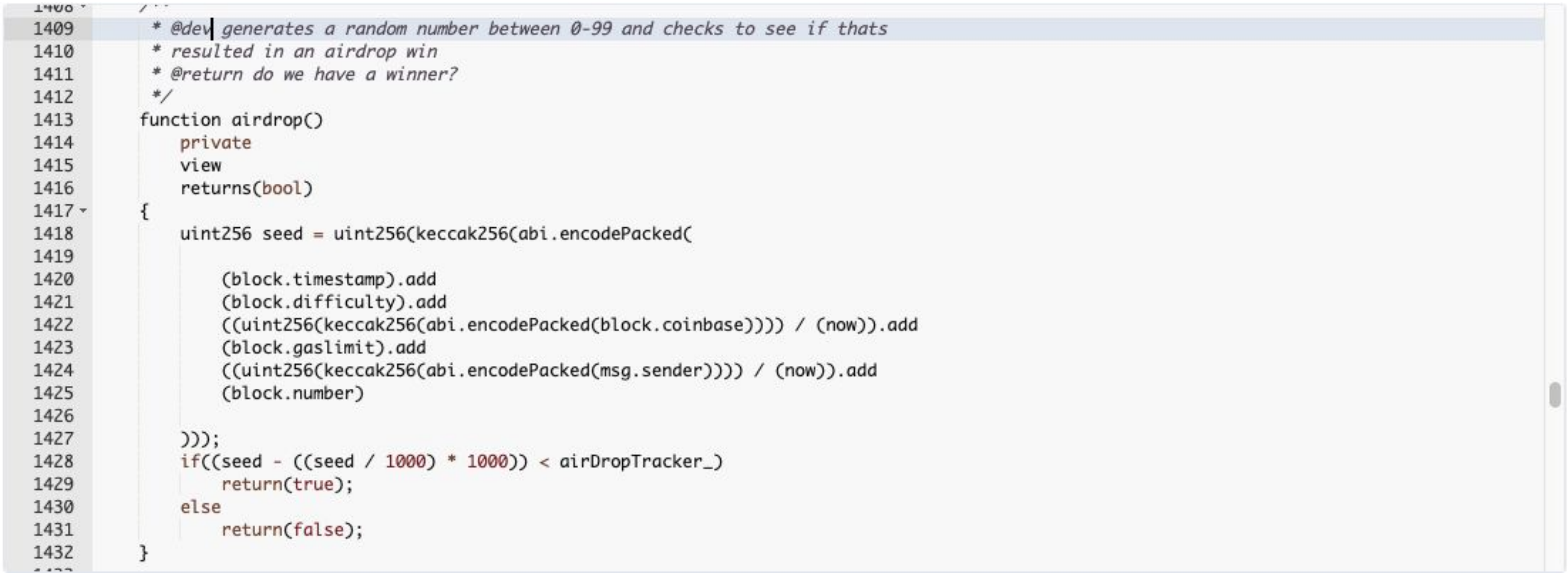
Random Airdrops

Fomo 3D - Random Airdrops

```
contract Fomo3D {

  uint256 pot;
  uint256 jackpot;

  function invest() payable {
    if (msg.value >= 0.1 Ether
        && airdrop()) {
      uint256 reward = pot.mul(75) / 100;
      msg.sender.send(reward);
    }

    // add a bit to the pot
    // add the rest to jackpot
    // extend the timer
  }

}
```

*** A simplified interpretation of an excerpt from Fomo3D*

Fomo 3D - Random Airdrops



```
     * @dev generates a random number between 0-99 and checks to see if thats
     * resulted in an airdrop win
     * @return do we have a winner?
     */
    function airdrop()
        private
        view
        returns(bool)
    {
        uint256 seed = uint256(keccak256(abi.encodePacked(

            (block.timestamp).add
            (block.difficulty).add
            ((uint256(keccak256(abi.encodePacked(block.coinbase)))) / (now)).add
            (block.gaslimit).add
            ((uint256(keccak256(abi.encodePacked(msg.sender)))) / (now)).add
            (block.number)

        )));
        if((seed - ((seed / 1000) * 1000)) < airDropTracker_)
            return(true);
        else
            return(false);
    }
```

*** An excerpt from Fomo3D*

Fomo 3D - Random Airdrops

There is no randomness on Ethereum!

All miners need to deterministically arrive to the same interpretation of the smart contract code.

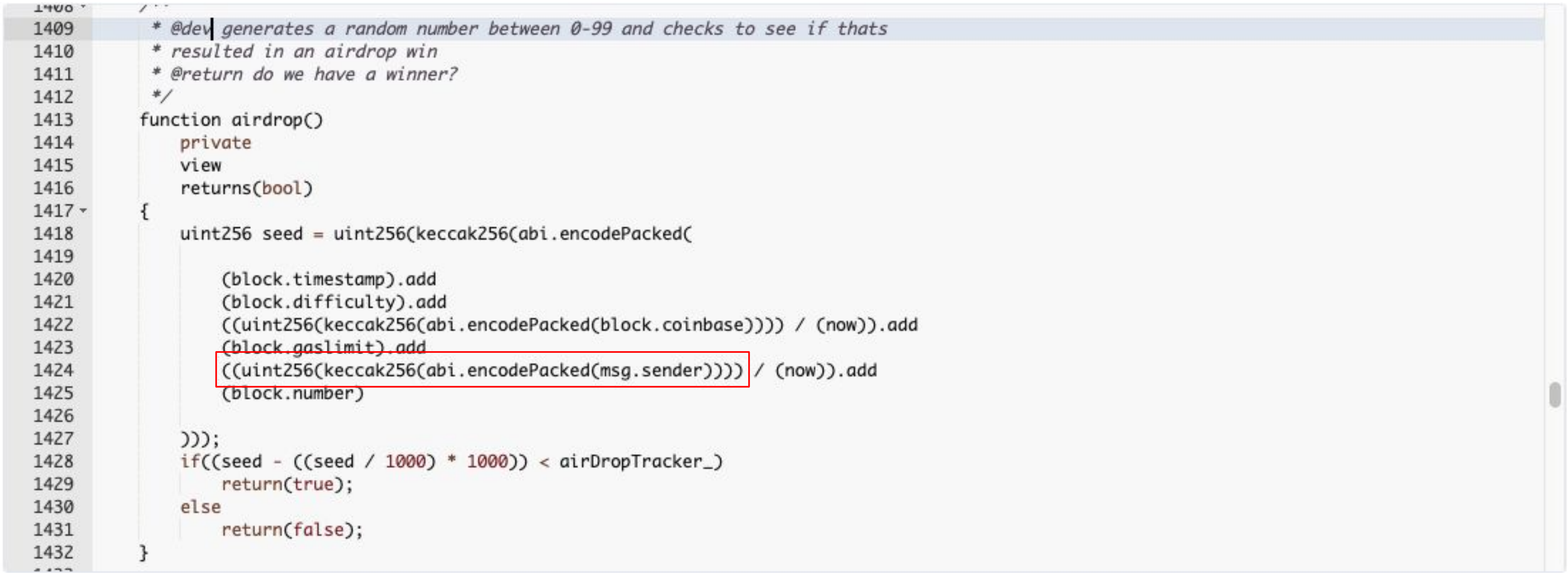
Fomo 3D - Random Airdrops Exploit

```
contract Exploit {

  function airdrop() { ... define exactly as in fomo3d ... }

  function exploit(address fomo3d) {
    if (this.airdrop()) {
      uint256 reward = fomo3d.pot().mul(75) / 100;
      if (reward > 0.1 Ether) {
        fomo3d.invest.value(0.1 Ether)();
      }
    }
  }

  // ...
  // more functions to make the whole thing useful

}
```

Fomo 3D - Random Airdrops

We can do even better...

Fomo 3D - Advanced Random Airdrops Exploit



```
     * @dev generates a random number between 0-99 and checks to see if thats
     * resulted in an airdrop win
     * @return do we have a winner?
     */
    function airdrop()
        private
        view
        returns(bool)
    {
        uint256 seed = uint256(keccak256(abi.encodePacked(

            (block.timestamp).add
            (block.difficulty).add
            ((uint256(keccak256(abi.encodePacked(block.coinbase)))) / (now)).add
            (block.gaslimit).add
            ((uint256(keccak256(abi.encodePacked(msg.sender)))) / (now)).add
            (block.number)

        )));
        if((seed - ((seed / 1000) * 1000)) < airDropTracker_)
            return(true);
        else
            return(false);
    }
```

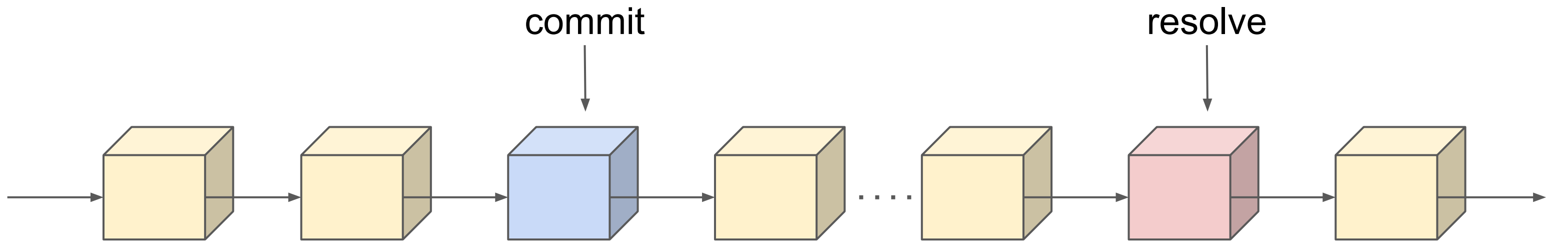
Fomo 3D - Random Airdrops Exploit

```
contract Exploiter { ... put the calling logic here ... }

contract JustDrain {

  function (address exploiter) { ... define exactly as in fomo3d ... }

  function exploit(address fomo3d) {
    Exploiter exploiter = new Exploiter();
    while (!this.airdrop(exploiter)) {
      exploiter = new Exploiter();
    }
    exploiter.callFomo3D();
  }

  // ...
  // more functions to make the whole thing useful

}
```

Fomo 3D - Random Airdrops

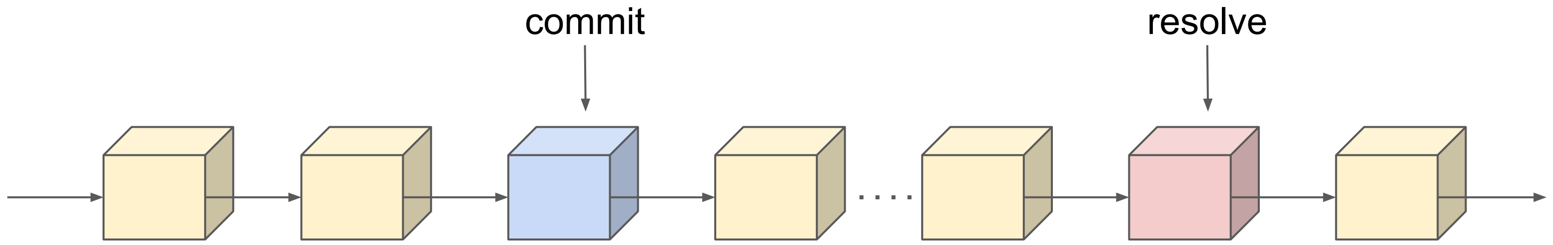
How to fix this?

Commit and Reveal



1. User commits to an action in block *n*
2. The action will be resolved *(n+k)* blocks later

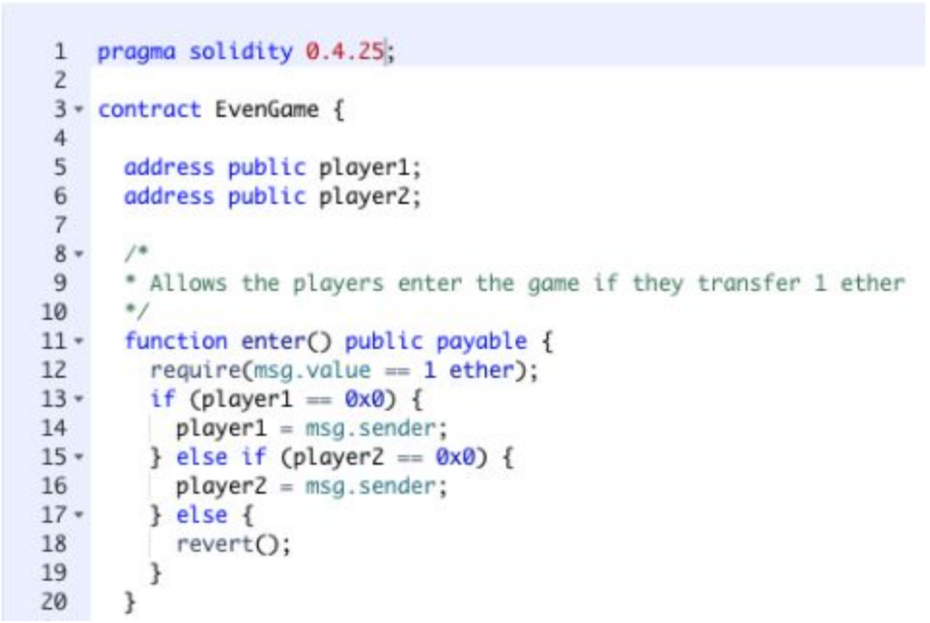
Commit and Reveal



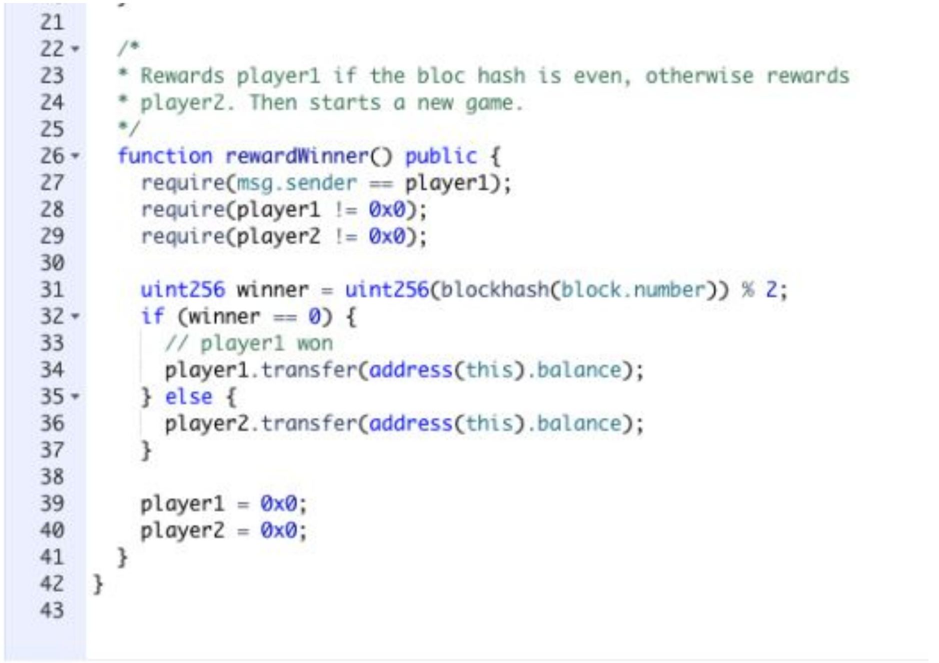
1. User commits to an action in block *n*
2. The action will be resolved *(n*+*k)* blocks later

The level of security is equivalent to the level of trust that the user does not have sufficient hashpower and incentive to mine *k* blocks ahead.

Naive Vulnerable Implementation

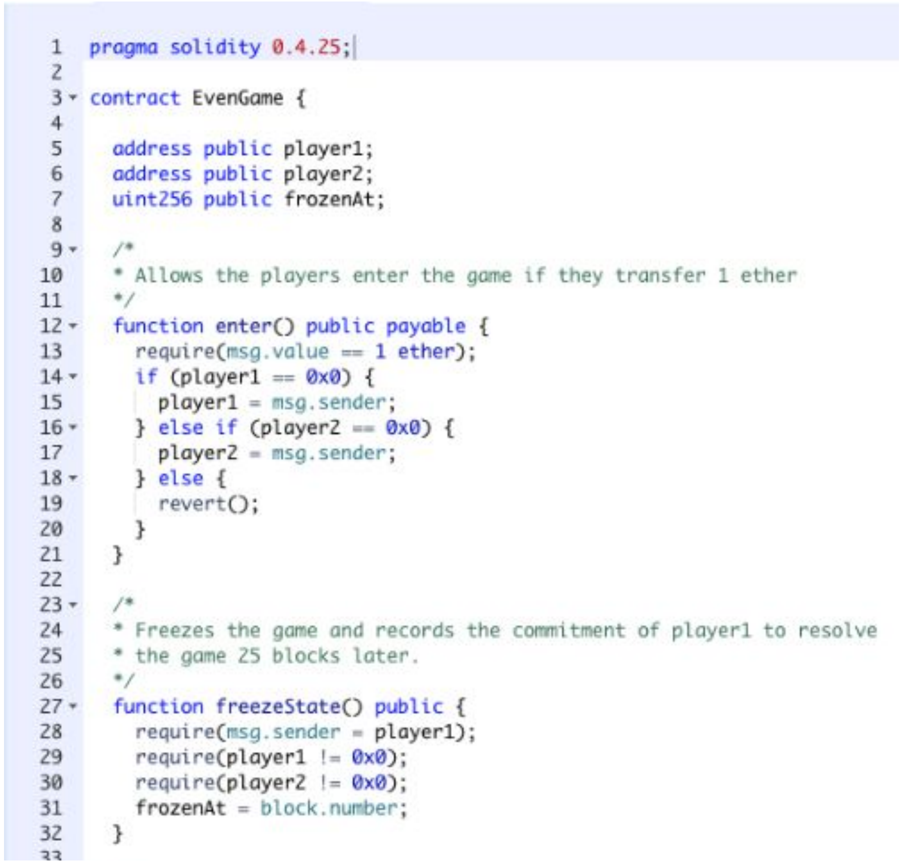


```
pragma solidity 0.4.25;

contract EvenGame {

  address public player1;
  address public player2;

  /*
  * Allows the players enter the game if they transfer 1 ether
  */
  function enter() public payable {
    require(msg.value == 1 ether);
    if (player1 == 0x0) {
      player1 = msg.sender;
    } else if (player2 == 0x0) {
      player2 = msg.sender;
    } else {
      revert();
    }
  }
```

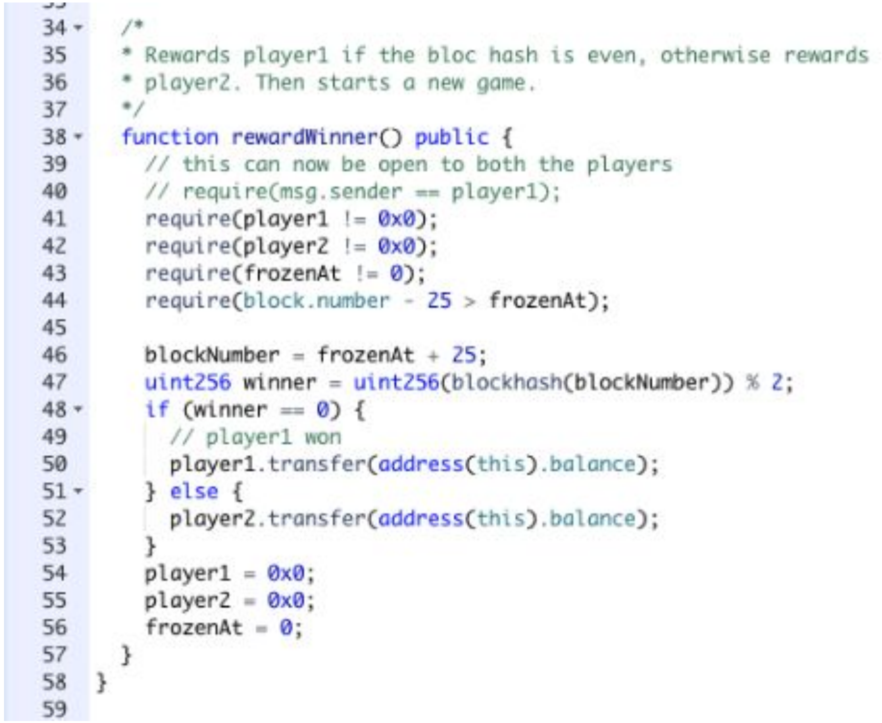


```

  /*
  * Rewards player1 if the bloc hash is even, otherwise rewards
  * player2. Then starts a new game.
  */
  function rewardWinner() public {
    require(msg.sender == player1);
    require(player1 != 0x0);
    require(player2 != 0x0);

    uint256 winner = uint256(blockhash(block.number)) % 2;
    if (winner == 0) {
      // player1 won
      player1.transfer(address(this).balance);
    } else {
      player2.transfer(address(this).balance);
    }

    player1 = 0x0;
    player2 = 0x0;
  }
}
```

Commit and Reveal Example



```
pragma solidity 0.4.25;

contract EvenGame {

  address public player1;
  address public player2;
  uint256 public frozenAt;

  /*
  * Allows the players enter the game if they transfer 1 ether
  */
  function enter() public payable {
    require(msg.value == 1 ether);
    if (player1 == 0x0) {
      player1 = msg.sender;
    } else if (player2 == 0x0) {
      player2 = msg.sender;
    } else {
      revert();
    }
  }

  /*
  * Freezes the game and records the commitment of player1 to resolve
  * the game 25 blocks later.
  */
  function freezeState() public {
    require(msg.sender = player1);
    require(player1 != 0x0);
    require(player2 != 0x0);
    frozenAt = block.number;
  }
```



```
  /*
  * Rewards player1 if the bloc hash is even, otherwise rewards
  * player2. Then starts a new game.
  */
  function rewardWinner() public {
    // this can now be open to both the players
    // require(msg.sender == player1);
    require(player1 != 0x0);
    require(player2 != 0x0);
    require(frozenAt != 0);
    require(block.number - 25 > frozenAt);

    blockNumber = frozenAt + 25;
    uint256 winner = uint256(blockhash(blockNumber)) % 2;
    if (winner == 0) {
      // player1 won
      player1.transfer(address(this).balance);
    } else {
      player2.transfer(address(this).balance);
    }
    player1 = 0x0;
    player2 = 0x0;
    frozenAt = 0;
  }
}
```

Fomo 3D

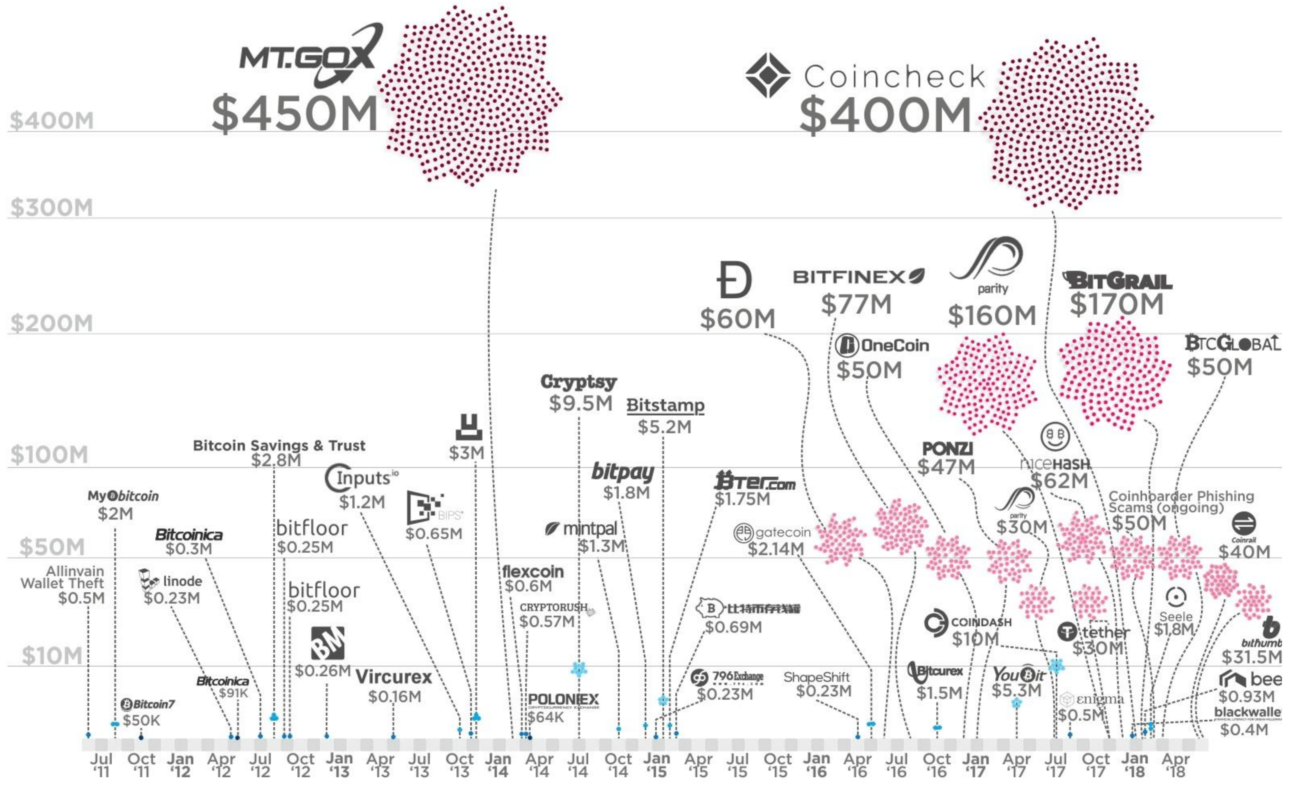
Fomo 3D: End of Game

Fomo 3D - Network Jamming

The first round of Fomo3D finished on August 22, 2018.

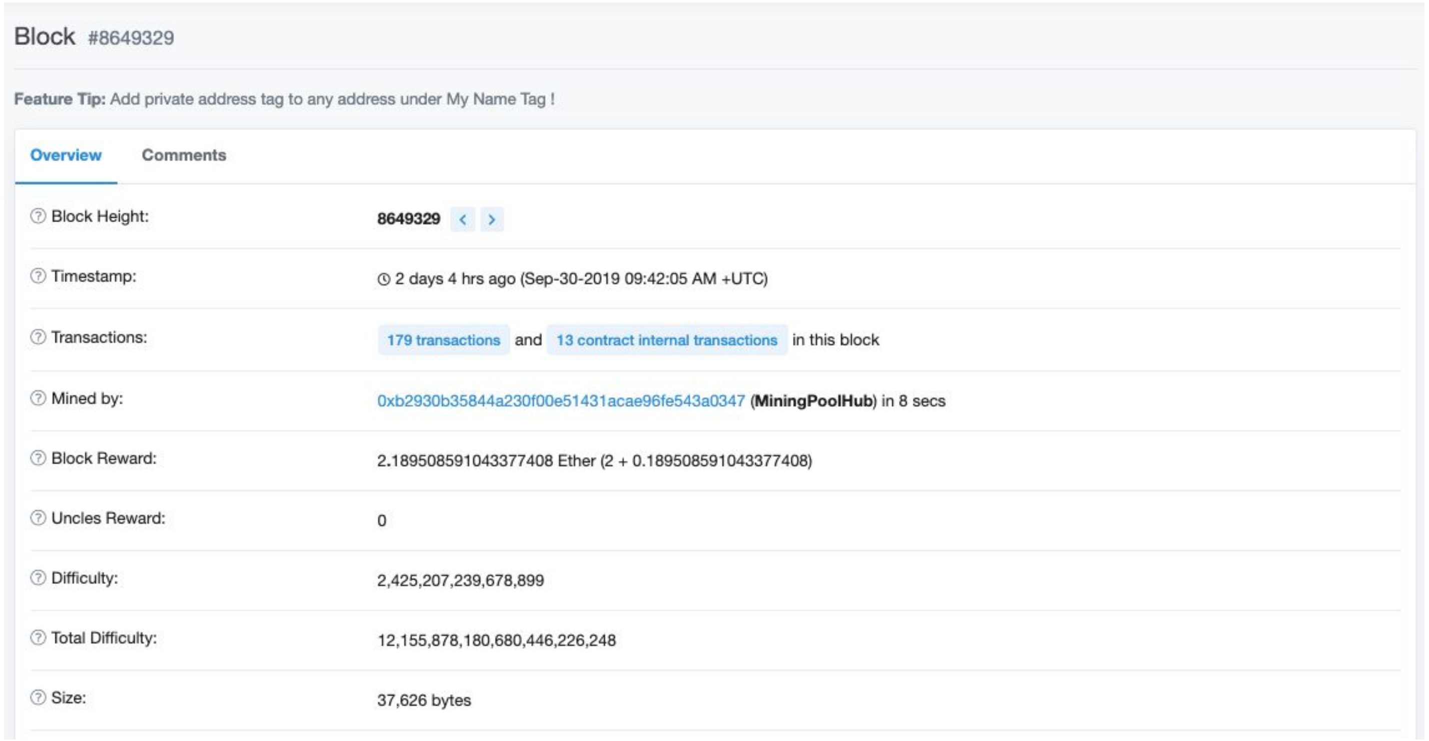
The winner collected 10,469 ETH (~$3.3 million @ $300 per Ether).

Fomo 3D - Network Jamming



https://howmuch.net/articles/biggest-crypto-hacks-scams

Fomo 3D - Regular Ethereum Block



Block #8649329

Feature Tip: Add private address tag to any address under My Name Tag !

Overview | Comments

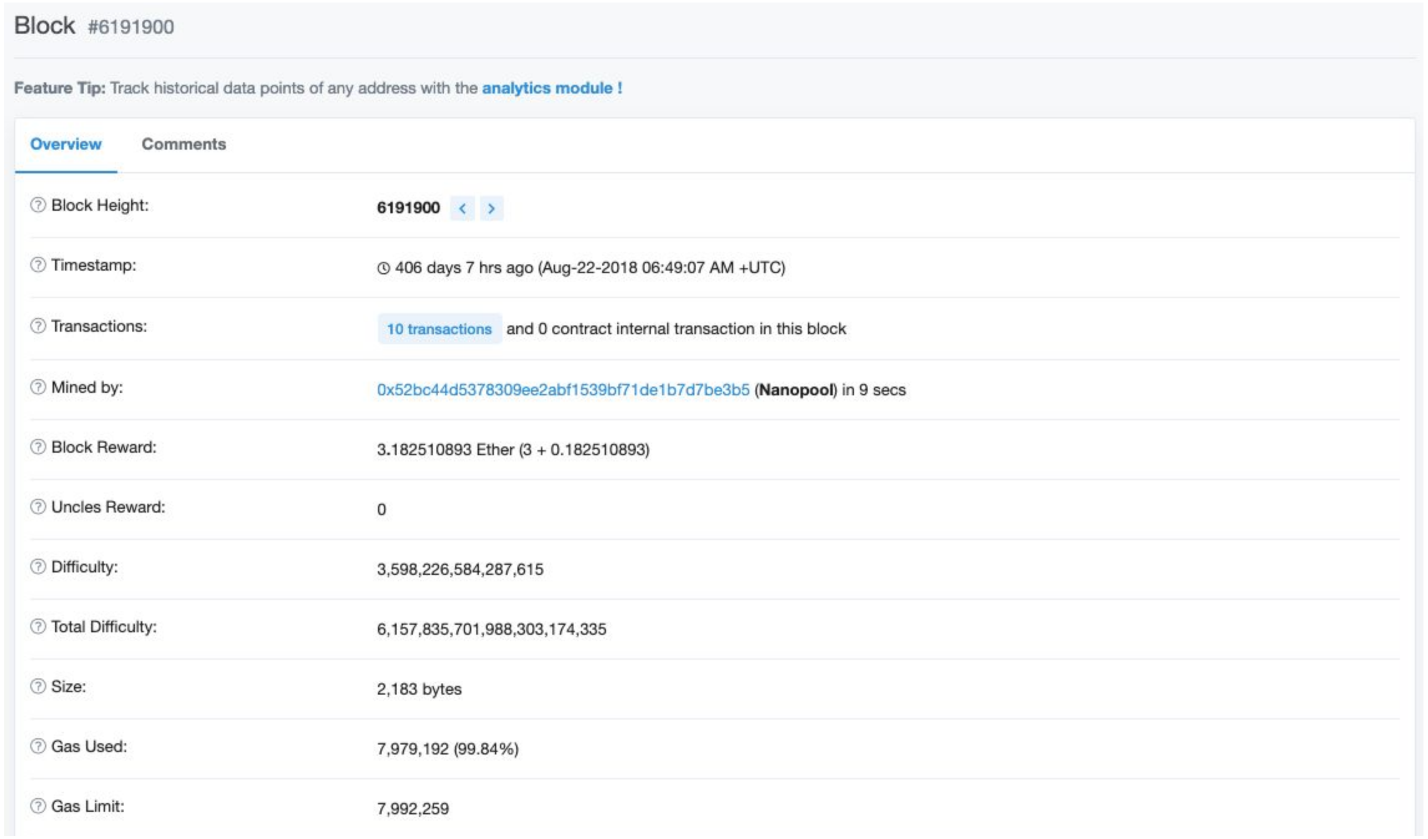
Block Height:	**8649329**
Timestamp:	2 days 4 hrs ago (Sep-30-2019 09:42:05 AM +UTC)
Transactions:	179 transactions and 13 contract internal transactions in this block
Mined by:	0xb2930b35844a230f00e51431acae96fe543a0347 (**MiningPoolHub**) in 8 secs
Block Reward:	2.189508591043377408 Ether (2 + 0.189508591043377408)
Uncles Reward:	0
Difficulty:	2,425,207,239,678,899
Total Difficulty:	12,155,878,180,680,446,226,248
Size:	37,626 bytes

Fomo 3D - Network Jamming

- Block 6191896:
  A user purchased a Fomo3D investment key.

- Block 6191898 - 6191906:
  Ethereum started showing abnormal number of transactions.

Fomo 3D - Jammed Block



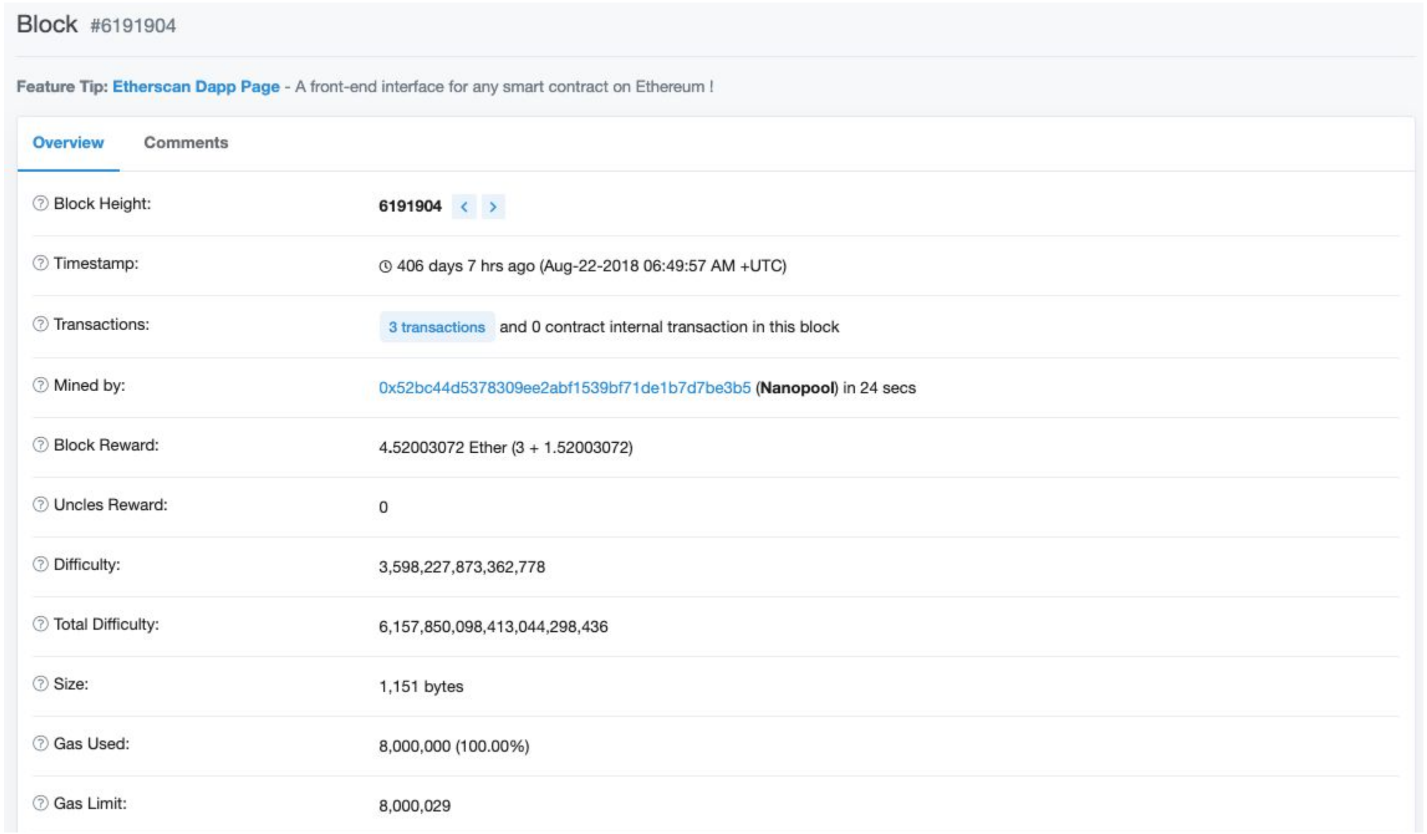
Block #6191900

Feature Tip: Track historical data points of any address with the **analytics module !**

Overview | Comments

Block Height:	**6191900**
Timestamp:	406 days 7 hrs ago (Aug-22-2018 06:49:07 AM +UTC)
Transactions:	10 transactions and 0 contract internal transaction in this block
Mined by:	0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5 (**Nanopool**) in 9 secs
Block Reward:	3.182510893 Ether (3 + 0.182510893)
Uncles Reward:	0
Difficulty:	3,598,226,584,287,615
Total Difficulty:	6,157,835,701,988,303,174,335
Size:	2,183 bytes
Gas Used:	7,979,192 (99.84%)
Gas Limit:	7,992,259

Fomo 3D - Jammed Block



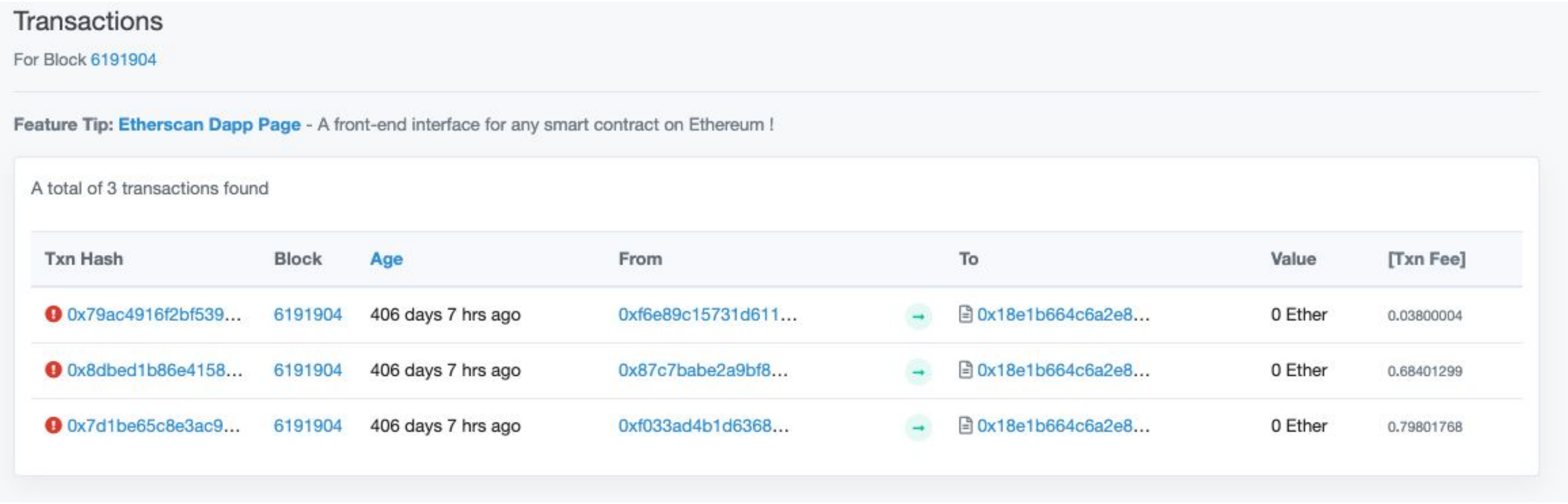
Block #6191904

Feature Tip: Etherscan Dapp Page - A front-end interface for any smart contract on Ethereum !

Overview Comments

Block Height:	**6191904**
Timestamp:	406 days 7 hrs ago (Aug-22-2018 06:49:57 AM +UTC)
Transactions:	3 transactions and 0 contract internal transaction in this block
Mined by:	0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5 (**Nanopool**) in 24 secs
Block Reward:	4.52003072 Ether (3 + 1.52003072)
Uncles Reward:	0
Difficulty:	3,598,227,873,362,778
Total Difficulty:	6,157,850,098,413,044,298,436
Size:	1,151 bytes
Gas Used:	8,000,000 (100.00%)
Gas Limit:	8,000,029

Fomo 3D - Jammed Block



Transactions

For Block 6191904

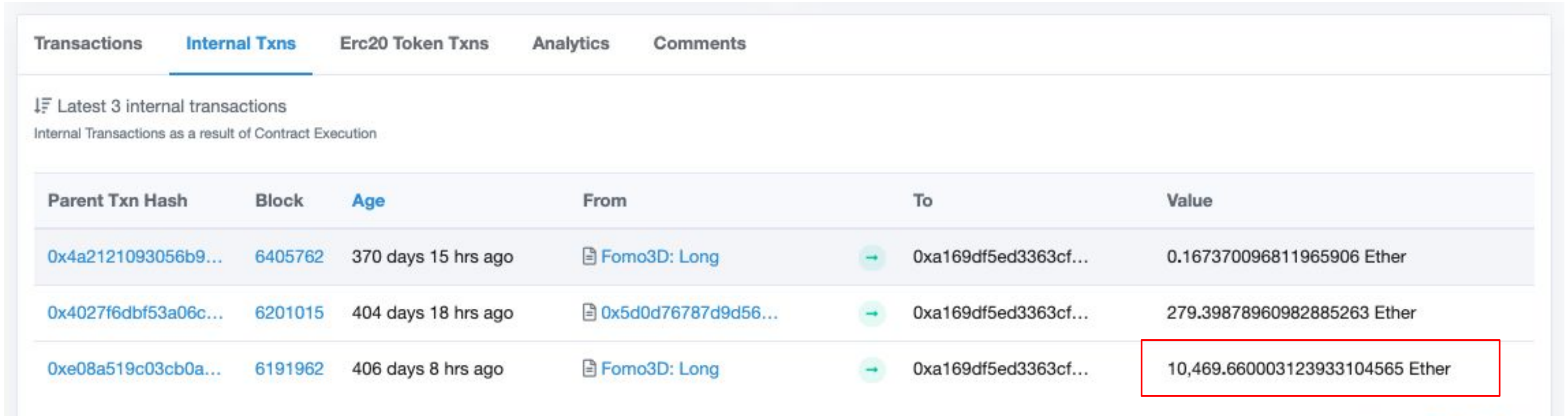
Feature Tip: Etherscan Dapp Page - A front-end interface for any smart contract on Ethereum !

A total of 3 transactions found

Txn Hash	Block	Age	From		To	Value	[Txn Fee]
❗ 0x79ac4916f2bf539...	6191904	406 days 7 hrs ago	0xf6e89c15731d611...	→	0x18e1b664c6a2e8...	0 Ether	0.03800004
❗ 0x8dbed1b86e4158...	6191904	406 days 7 hrs ago	0x87c7babe2a9bf8...	→	0x18e1b664c6a2e8...	0 Ether	0.68401299
❗ 0x7d1be65c8e3ac9...	6191904	406 days 7 hrs ago	0xf033ad4b1d6368...	→	0x18e1b664c6a2e8...	0 Ether	0.79801768

Fomo 3D - Network Jamming

- Block 6191896: A user purchased a Fomo3D investment key.
- Block 6191898 - 6191906: Ethereum started showing abnormal number of transactions.
- This time was sufficient for the limit for the next key purchase to lapse.



Latest 3 internal transactions

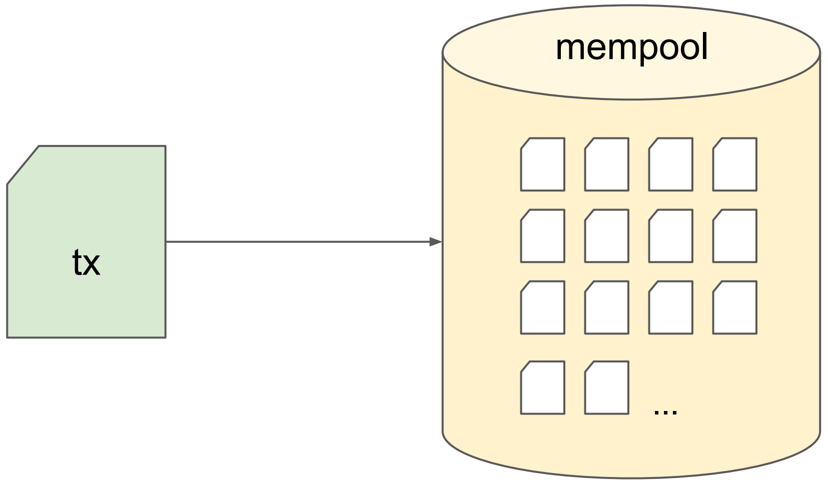
Internal Transactions as a result of Contract Execution

Parent Txn Hash	Block	Age	From		To	Value
0x4a2121093056b9...	6405762	370 days 15 hrs ago	Fomo3D: Long	→	0xa169df5ed3363cf...	0.167370096811965906 Ether
0x4027f6dbf53a06c...	6201015	404 days 18 hrs ago	0x5d0d76787d9d56...	→	0xa169df5ed3363cf...	279.39878960982885263 Ether
0xe08a519c03cb0a...	6191962	406 days 8 hrs ago	Fomo3D: Long	→	0xa169df5ed3363cf...	10,469.660003123933104565 Ether

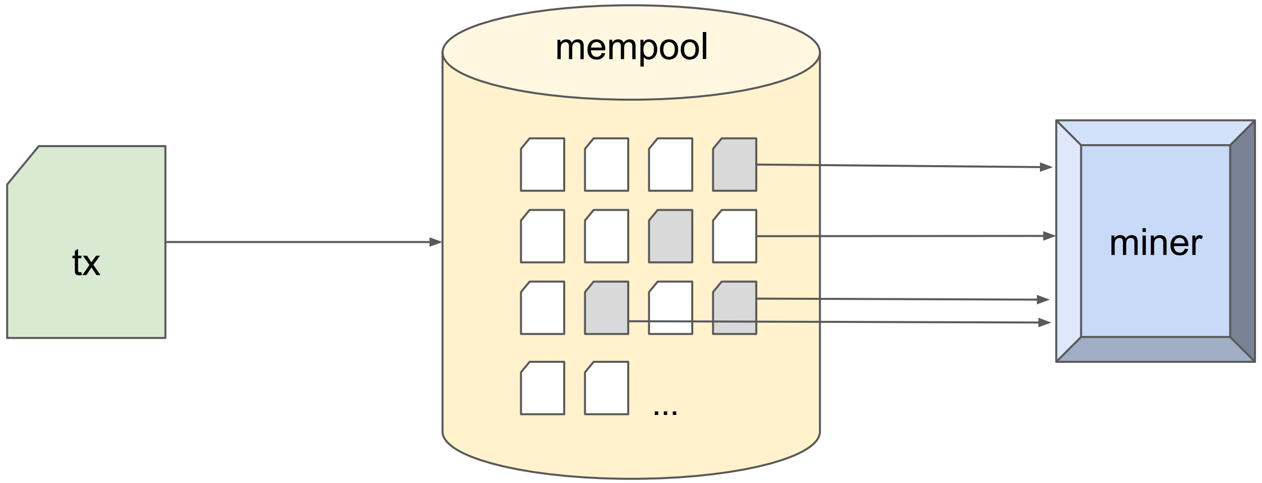
Fomo 3D - Network Jamming

How can such a thing happen?

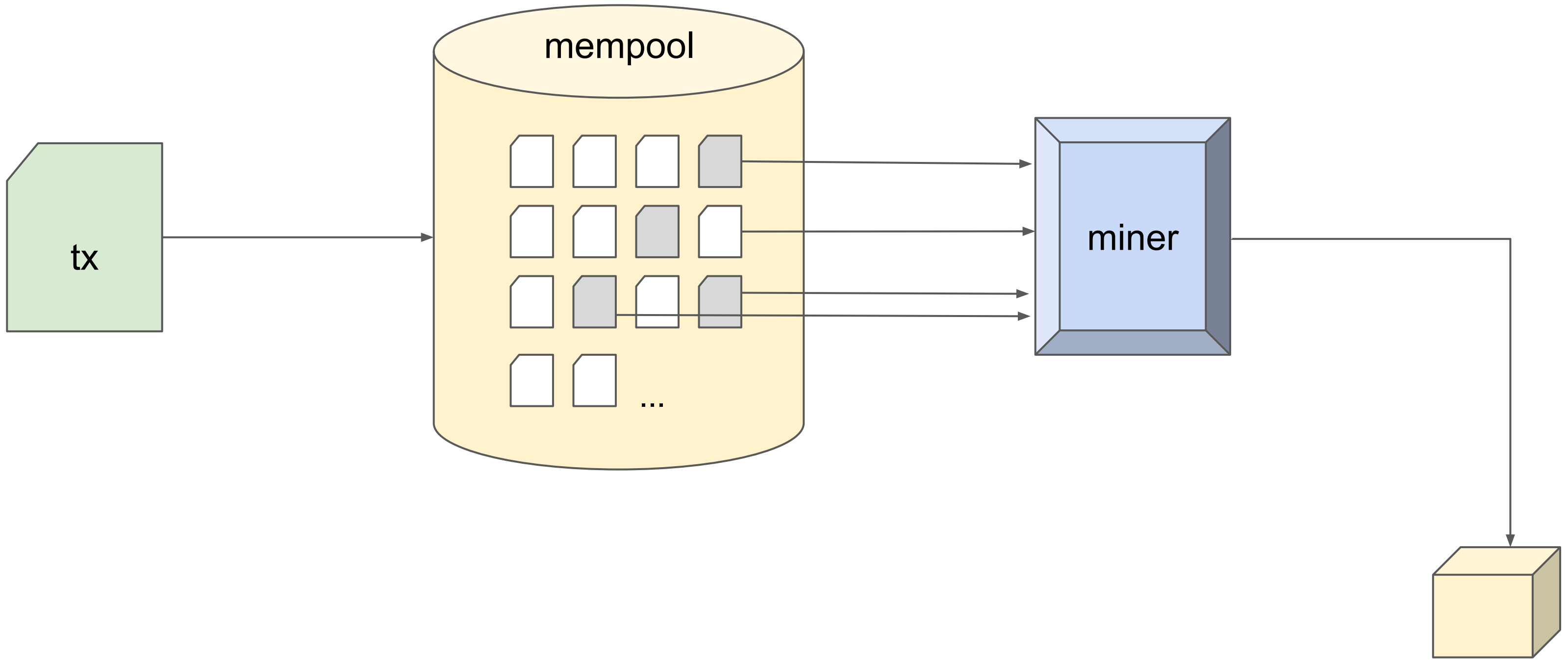
How Did This Work: Transaction Mining



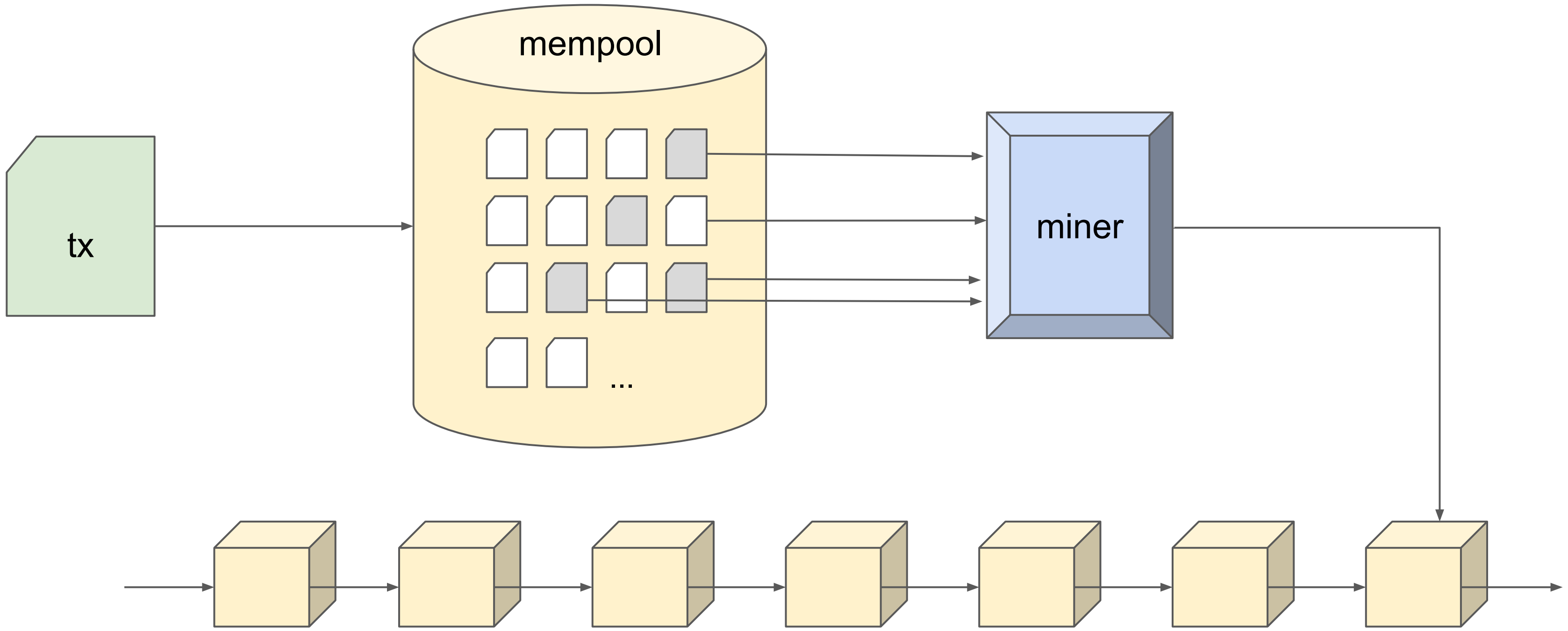
How Did This Work: Transaction Mining



How Did This Work: Transaction Mining

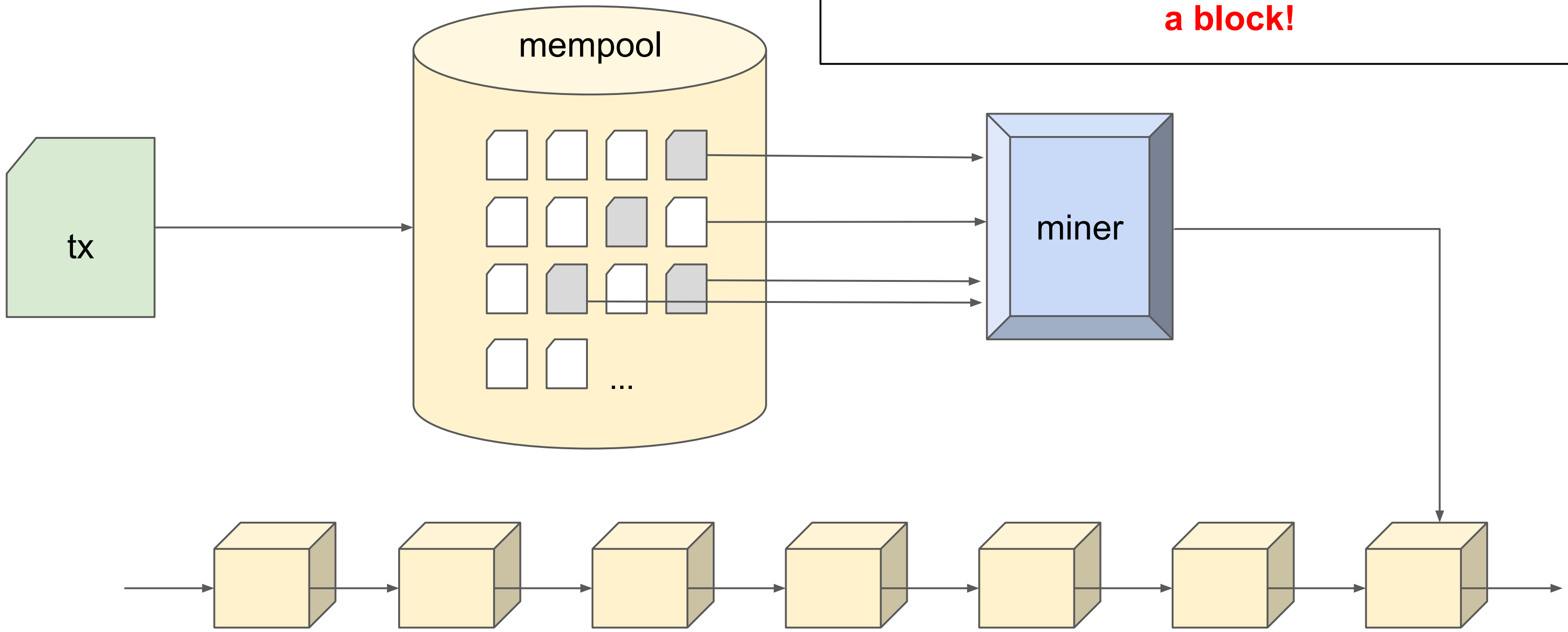


How Did This Work: Transaction Mining



How Did This Work: Transaction Mining

Miner chooses transactions to include in a block!

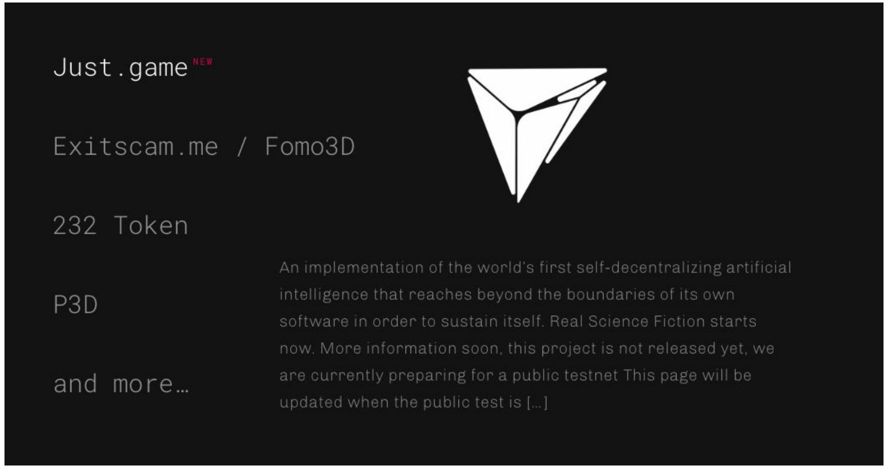


Fomo 3D - Final Remarks

Final Remarks

- Someone is playing again: Fomo3D holds ~1,253.00 Ether
- User interface (www.exitscam.me) does not seem to be active

Fomo 3D - Final Remarks


Just.game NEW

Exitscam.me / Fomo3D

232 Token

P3D

and more…

An implementation of the world's first self-decentralizing artificial intelligence that reaches beyond the boundaries of its own software in order to sustain itself. Real Science Fiction starts now. More information soon, this project is not released yet, we are currently preparing for a public testnet This page will be updated when the public test is [...]

An implementation of the world's first self-decentralizing artificial intelligence that reaches beyond the boundaries of its own software in order to sustain itself.

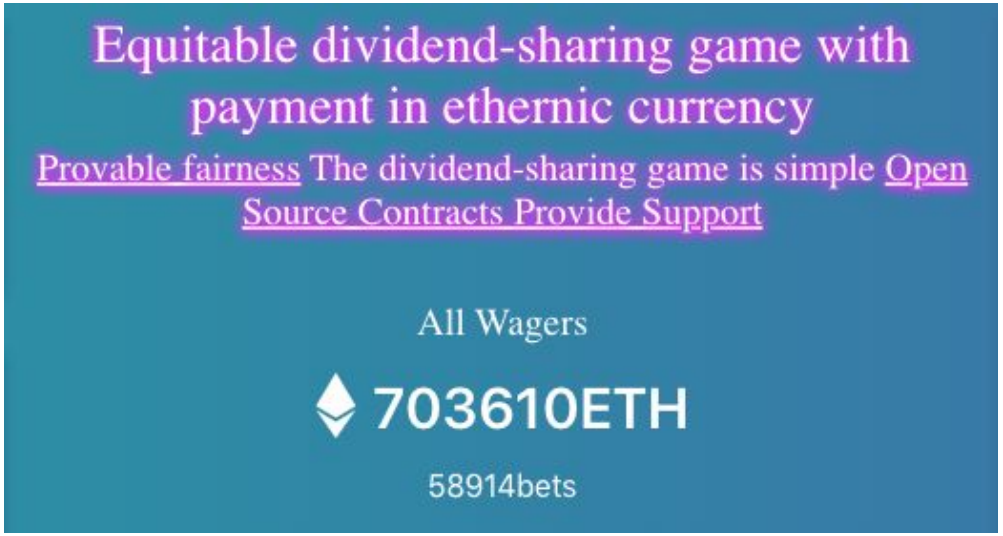
FairWin

FAIRWIN.ME
Marshall
initiators of the FAIRWIN
FAIRWIN Public Winning Intelligent Contract Game
FAIRWIN公赢智能合约游戏

FairWin

FairWin

FairWin: A ponzi scheme deployed on July 27, 2019. On September 26, it held ~49,518 ETH. Vulnerabilities were found and reported in September 2019. The contract is currently empty.


Equitable dividend-sharing game with payment in ethernic currency

Provable fairness The dividend-sharing game is simple Open Source Contracts Provide Support

All Wagers

703610ETH

58914bets

FairWin

Throughout late September 2019, the community found disclosed two vulnerabilities:

1. **Centralization of Power** allowing the owner of the contract to drain all the invested funds
2. **Front running** vulnerability allowing anyone to compete for another person's investments

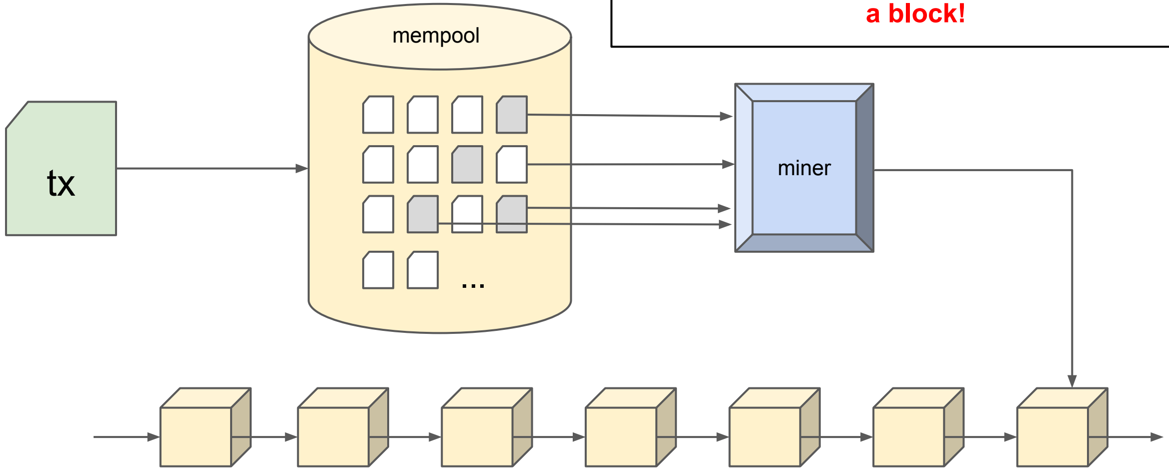
Front Running

Front Running: An attempt to get one transaction to be mined before another.

Front Running

Front Running: An attempt to get one transaction to be mined before another. This is often done through offering excessive gas prices.

Miner chooses transactions to include in a block!



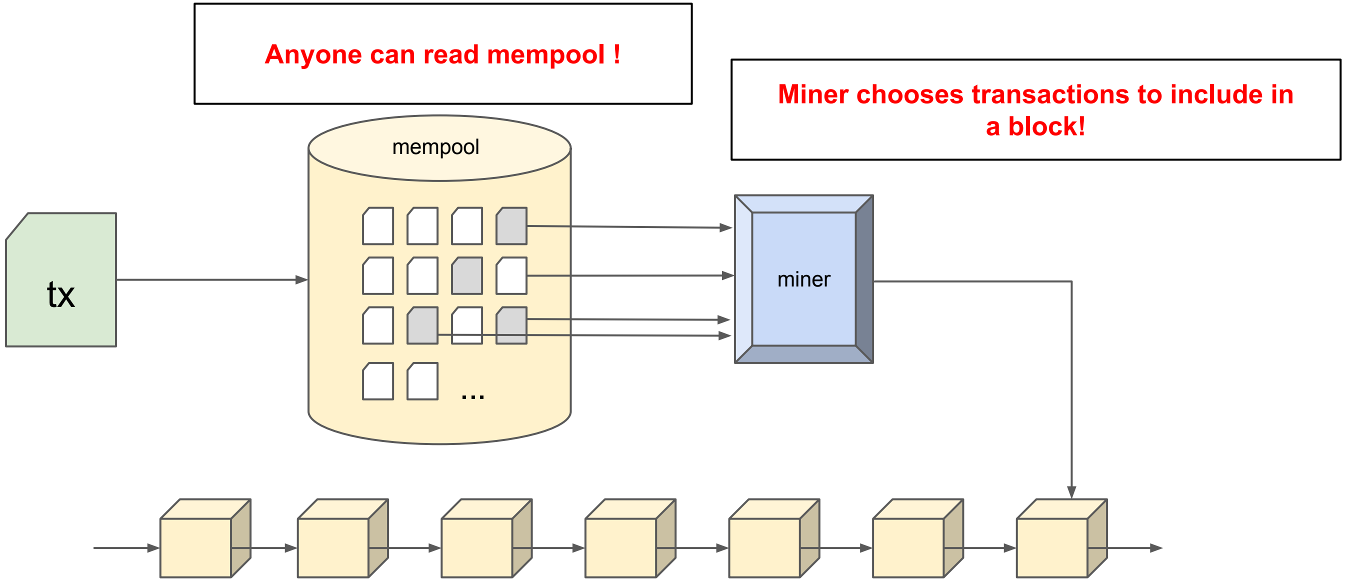
FairWin

```
contract FairWin {

  mapping (string => address) investors;
  mapping (string => uint256) investments;

  function invest(string inviteCode, ...) payable {
    if (investors[inviteCode] == 0x0) investors[inviteCode] = address;
    investments[inviteCode] = investments[inviteCode].add(msg.value);

    //… other logic
  }

  function exit(string inviteCode) {
    require(investors[inviteCode] == msg.sender);
    investors[inviteCode].send(investments[inviteCode]);
  }

}
```

*** An illustration of a vulnerability similar to FairWin*

Mempool Problem

Front Running: An attempt to get one transaction to be mined before another. This is often done through offering excessive gas prices.


Anyone can read mempool !
Miner chooses transactions to include in a block!
mempool
tx
miner
...

Mempool Problem

One more example...

Escrow Example

```
contract Escrow {
  mapping (bytes32 => uint256) deposits;
  mapping (bytes32 => bool) invalid;

  function deposit(bytes32 secret) payable {
    require(!invalid[secret]);
    deposits[secret] = deposits[secret].add(msg.value);
  }

  function withdraw(string password) {
    bytes32 secret = keccak256(password);
    uint256 value = deposits[secret];

    require(value > 0);
    deposits[secret] = 0;
    invalid[secret] = true;
    msg.sender.send(value);
  }
}
```

Combination

Can we combine the two techniques?

Effective Front Running

1. **Front Running**
    a. Read the mempool
    b. Submit competing transactions with higher gas price

2. **Jam the network**
    a. Submit transactions exhausting block gas limit
    b. Nobody’s transactions will be mined

3. **Jam the network + Front Running**
    a. Submit transactions exhausting block gas limit
    b. Nobody’s transactions will be mined, but they **will be present in mempool**
    c. Read the mempool
    d. Submit competing transactions with higher gas price
    e. Unblock the network

Thank you!

Questions?

Quantstamp™

martin@quantstamp.com