

The first Flashbots bundle was mined in block 11550019 on Dec-29-2020 04:33:22 PM UTC, kicking off what is poised to be the year of MEV.

We can't help but notice parallels between the \$GME debacle that is causing increased scrutiny in the role of privileged stock market participants, and what we've described as the MEV crisis. Averting this impending crisis is in our eyes essential to avoid perpetuating the same flaws in DeFi that we've seen traditional financial markets structure suffer from.

This post is our first monthly update on the activity happening on Flashbots, as part of our ongoing efforts in fulfilling Flashbots' commitment to iterate in the public. In addition to open-sourcing our codebase, and providing periodic progress updates via our MEV Roasts, this update aims to review the searcher and miner activity on Flashbots core infrastructure.

## Overview

One month ago, on January 6, we soft-launched the Flashbots Alpha on our discord. It provides a market-based alternative to unilateral trader-miner collusion for the extraction of MEV. Anyone with an internet connection can compete in a fairer and more efficient MEV extraction game by writing a searcher bot that submits Flashbots transaction bundles with a tip to a set of participating miners running MEV-geth through MEV-relay, without the need to strike bespoke deals with any miners.

Over the course of January, our engineering efforts were primarily focused on improving MEV searcher tooling and improving our relay scalability. We hope to soon be able to say goodbye to API-keys and remove rate limits. We have also embarked on longer-term research on potential designs of Flashbots core infrastructure with full privacy.

## Searcher Adoption

Flashbots does not actively participate in proprietary MEV search activities. Our efforts are currently centered around providing open-source example implementations and toolings for third-party MEV searchers within the Flashbots community.

### Open-source MEV searcher template

In January, we open sourced the simple-arbitrage searcher which provides a template for how to write a Flashbots searcher bot. Simple-arbitrage is a bot which inspects Uniswap and Sushiswap market pairs for 2-legged arbitrage opportunities, submits them to Flashbots for execution, and tips the miner for being included at the top of the block they produced. We're happy to see community searchers successfully deploy and modify this simple-arbitrage searcher bot to compete for arbitrage opportunities on Flashbots.

### MEV searcher tooling

Based on our initial survey of requirements from traders and bot operators, we are designing interfaces and feature-sets that plug into popular Ethereum libraries used in bot operation to provide developers with commonly needed functionalities when submitting to `eth_sendBundle`. In January we worked on: \* Integration with the `ethers.js` library: contains the `FlashbotsBundleProvider` `ethers.js` provider to provide high-level access to `eth_sendBundle` rpc endpoint. \* Integration with the `web3.py` library which will be released soon.

Another useful searcher tooling we are working on is a new Ethereum JSON-RPC endpoint to allow searchers to simulate a Flashbots bundle at the head of a block, where Flashbots bundles will always land based on the current implementation of MEV-geth.

## Whitehat rescue of NFTs via Flashbots

One of our core devs executed a whitehat rescue of over \$150k with Samczsun, for an account with a compromised private key. While the account funds were rescued without Flashbots, we used Flashbots to rescue the account's remaining NFTs, including CryptoKitties, Axies and Gods Unchained cards.

As far as we know, this was the first NFT rescue on Ethereum, uniquely enabled by Flashbots's ability to separate the block.coinbase tip to the miner from the other txs in a Flashbots transaction bundle. The whitehats cleared the compromised account of all ETH and set up an ETH sweeper in order to stop the hacker from moving the NFTs away from the account, as the hacker would have needed ETH to pay the related tx fee. They then executed a Flashbots bundle of transactions with the relevant NFT-saving txs, and successfully rescued the NFTs without leaking their strategy.

We recorded a live demo of this rescue operation using Flashbots during the Jan 13 edition of MEV Roast. We are happy to see one of our Discord community members applying the same method using Flashbots to create a sanctuary for lost ENS.

We are still in the process of open-sourcing the white hat rescue creation framework. We would like to encourage whitehat security researchers to uncover more creative use cases of Flashbots for good.

## MEV search activities on Flashbots

Over the course of January, a total of 19 searcher bots have successfully landed a total of 1485 bundles on chain and paid a total of 51.76 ETH in fees to miners.

□  
□

There were a wide range of MEV strategies landed on chain through Flashbots — including some fairly innovative ones, such as an incentivized call, and market-taker that arbs without causing user harm.

- NFT rescue — Axie
- incentivized call
- flashloan arbitrage
- hello world
- market taker

We are working on a public dashboard that will allow you to keep track of these aggregate metrics. For the time being, you can keep up with on-chain Flashbots activity by following @epheph on twitter for 'MEV of the Week' analysis, @HighGwei for alerts on big Flashbots miner tips, and the #bundle-alerts channel in Flashbots Discord server for a stream of Flashbots transactions as they are mined.

## Miner Participation

### Flashbots miner support

Flashbots provides technical support to miners and mining pools adopting MEV-geth during Alpha testing. Amongst the mining pools who indicated interest to test Flashbots, UUpool has been the first pool to set up and run MEV-geth in production. They've collected virtually all of the additional miner revenue from MEV searchers' block.coinbase tips for transaction inclusion through Flashbots during the month of January.

### Blocks mined including Flashbots transaction bundles

Below is a graph of the % of total blocks mined by a Flashbots-enabled miner. It shows that over the course of January, MEV-geth consistently received between 3–4% of the total network hashrate.

□

The following graph shows all the blocks mined by Flashbots miners over the course of the month. The breakdown of the blocks mined by the participating mining pool shows how frequently Flashbots blocks (green), which are blocks mined with Flashbots transaction bundles, are more valuable than regular blocks (blue).

□

### Observed deviation from expected miner behavior and subsequent correction

While MEV-geth has a built-in profit switcher which guarantees that the most valuable block is mined, we noticed a deviation from the expected behavior where the miner sometimes mined a Flashbots block when a regular block would have been more profitable and vice-versa. Our investigation revealed that the participating mining pool disabled the profit switcher in MEV-geth and was instead using a proprietary profit switcher based on their custom mining pool setup which ended up leaving money on the table. We notified the mining pool of the suboptimal outcome of their set-up and saw a subsequent correction towards the end of the month.

We recommend miners to use the built-in profit switcher since using third party or custom profit switchers risks negatively impacting their profitability. That being said, mining pools often run on custom infrastructure which does not always make it possible to use the built-in profit switcher. We are actively looking for suggestions on how to improve mev-geth and make it easier to integrate with complex mining pool infrastructures without loss of profit.

An additional insight to glean from the graph above is the presence of "other" blocks (yellow). These represent blocks that follow neither the regular transaction order by gas price nor the Flashbots transaction order. Our investigation revealed that UUpool was testing either MEV extraction or direct collusion by including a series of sandwich arbitrages outside the Flashbots system. This resulted in a high frequency of blocks with MEV extraction with no payment to the miner on-chain.

Many Flashbots community members raised concerns about this direct inclusion bypassing MEV searcher competition and the impact it has on the fairness of the system. We were able to quickly rule out the hypothesis that UUpool was front-

running Flashbots bundles by comparing the direct inclusion transactions to the transaction bundles sent through the Flashbots MEV-relay. [Going forward, we are considering providing public access to the content of historical bundles and would like community feedback to inform our decision.]We brought the community concern over fairness of competition to the attention of the participating mining pool. Towards the end of the month, we observed the pool had stopped including transactions without miner tips. So all MEV transaction bundles now compete on the same playing field by competing in the seal-bid auction .

## **Establishing norms for a healthy ecosystem**

Although no miner front-running of Flashbots transaction bundles was observed, and subsequent adjustment has been made by participating miners, we caution miners against engaging in direct MEV collusion during the Flashbots Alpha as it erodes the fairness and efficiency of the system as well as prevents the Flashbots core devs from gathering essential data on the health of the system. Miners exposing their endpoint for direct collusion with searchers allow MEV searchers to bypass the sealed bid auction mechanism of MEV-Geth. From a miner's perspective, this reduces the potential revenue from a competitive MEV-extraction ecosystem and opens them up to potential DoS attacks.

In the long term, the Flashbots roadmap addresses community concerns on fairness and health of the Flashbots system by technically guaranteeing full privacy. We would like to invite privacy researchers and interested community members to review our architecture and contribute to improving the Flashbots architecture design