

On Wednesday August 9, 2023 at 11:55pm UTC, one of Flashbots' builders began erroneously including reverted transactions in blocks. Even though these transactions were not allowed to revert (eg. they were not included in the [revertingTxHashes

list](https://docs.flashbots.net/flashbots-auction/searchers/advanced/rpc-endpoint#eth\_sendbundle)), their associated bundles were still included on chain.

Over a period of ~2 hours, the affected Flashbots builder landed 722 reverted transactions that were not intended to be included on chain. Users lost approximately 10.2261 ETH in gas fees as a result.

Although only one builder was affected, Flashbots paused operation of all builders out of an abundance of caution for ~4 hours to investigate the issue and identify a stable release to roll back to. During this window, no transactions submitted to Flashbots Protect or the Bundle Relay were landed by Flashbots builders. Flashbots gradually restarted each builder after extensive testing.

## Timeline

- [11:55pm UTC] Flashbots builder 0xb89b93

lands the [first incorrectly-reverting transaction](#)

- [12:34am UTC] A community member reports misbehavior of the Flashbots builder in [discord](#)
- [1:29am UTC] Another community member tags the Flashbots team in a subsequent [discord report](#) and the team becomes aware of the issue
- [2:30am UTC] Flashbots shuts down all builders and [notifies](#) the community of the outage on discord
- [2:55am UTC] Flashbots attempts to restart seemingly-unaffected builder 0x81babe
- [3:15am UTC] Flashbots shuts down builder 0x81babe

out of abundance of caution due to what appeared to be bad blocks (though after further investigation, Flashbots confirmed builder 0x81babe

was operating normally)

- [4:40am UTC] Flashbots informs top Protect RPC users of outage via telegram
- [5:00am UTC] Flashbots stops sending bundles to SGX builder 0xa35e2b
- [6:50am UTC] Flashbots restarts builder 0x81beef

in dry run, where it produces but does not submit blocks. Flashbots manually confirms that reverted transactions are not incorrectly included in any blocks produced by builder 0x81beef.

- [6:58am UTC] Flashbots restarts builder 0x81beef
- [5:19pm UTC] Flashbots resumes sending bundles to SGX builder 0xa35e2b
- [August 11] Flashbots tests and restarts all other unaffected builders
- [August 14] Flashbots restarts affected builder 0xb89b93

## Root cause

Flashbots runs several builders that receive bundles. Analysis confirmed that all blocks which contained incorrectly-reverting transactions were produced by builder with pubkey 0xb89b9308fbc6c2998c7e60e39424b858c74b02c234b3e0fa5ecf7c3971208dfa5f92e0bdbe16fc24abfd71c248acf0f9

between block [17880755](#) and block [17881274](#). No other builders demonstrated this behavior.

The incident occurred because Flashbots deployed a code change to builder 0xb89b93

on August 9— approximately the same time as the [first incorrectly-reverting transaction](#) was landed. This code change was related to performance optimizations intended to improve the efficiency of the bundle merging process. However, the deployed change mistakenly did not check the revert status of submitted transactions. This bug was missed and made its way into production despite being reviewed by 2 engineers.

# Mitigations

Flashbots restarted the unaffected builders after identifying that they did not land any bundles with incorrectly-reverting transactions. Flashbots then reverted the change that had been deployed to the affected builder, rolled it back to a stable release, and monitored its performance locally. Flashbots did not observe any incorrectly-reverting transactions for the following 3 days. Flashbots then restarted the affected builder on the stable release.

To prevent similar issues from arising again, Flashbots has added new checks in the builder and monitoring to detect if bundles are included incorrectly. No builder has triggered alerts since the rollback and restart. Going forward, Flashbots is also investing in improving how deployments are done to make our testing and review process more robust. This involves testing for key invariants in dry run, which provides a line of defense for when human-led reviews fail.

## Sources

Statistics on incorrectly-included transactions are pulled from a combination of Flashbots bundle data, Infura, and [mempool.guru](https://mempool.guru).