On June 10th 2024, the ultra sound relay accepted an impossibly high bid value (1e41 ETH) in slot 9265720. That is, a block claiming to pay more ETH than exists on the network in total managed to pass simulation.

On further investigation, we found that this was caused by an underflow bug in the reference implementation of the code that MEV-Boost relays use to simulate and validate builder's blocks. The underflow occured when the proposer balance decreased by more than the builder payment in a block.

In theory, a builder could abuse this bug by waiting for a block with this property, and posting an arbitrarily high bid but extremely low payment transaction. This would allow the builder to win the block (because their bid seems extremely high) while paying the proposer next to nothing (because their actual payment is extremely low). In practice, we do not see evidence that this was systematically exploited. At most, the bug would have resulted in the proposer for slot 9265720 receiving a slightly lower value block.

Ultra sound, Bloxroute, Agnostic, Aestus, Titan, and Flashbots turned off headers on their MEV-Boost relays for approximately 2 hours out of an abundance of caution to prevent incidents and patch the issue. After applying the patch, ultra sound and other relays have seen no evidence of impossibly high bids or error messages related to the underflow bug. Flashbots is upstreaming the audited patch into the reference implementation of the payload validation service.

Timeline

- June 10th, 2024 at 2:13pm UTC: Ultra sound relay reports underflow
- 2:23pm UTC: Incident response group established with MEV-Boost relay teams
- 2:30pm UTC: Bug confirmed
- 2:50pm UTC: Flashbots relay shuts off headers
- 2:56pm UTC: Ultra sound and Aestus shut off headers
- 2:59pm UTC: Bloxroute shuts off headers
- 3:13pm UTC: Flashbots creates apatch and reviews with other relay teams
- 4:02pm UTC: Agnostic disables builder.validation_use_balance_diff
- 4:04pm UTC: Flashbots confirms patch passes unit test
- 4:09pm UTC: Ultra sound deploys patch to holesky and confirms that errors related to the bug no longer appear in logs
- 4:33pm UTC: Ultra sound deploys patch on mainnet and monitors for errors in 100k simulations (finding none)
- 4:35pm UTC: Aestus deploys patch on mainnet
- 5:15pm UTC: Eden deploys patch on mainnet
- 5:18pm UTC: Flashbots relay lands ablock with the patch applied
- 5:37pm UTC: Titan deploys patch on mainnet
- 5:38pm UTC: bloXroute deploys patch on mainnet
- 5:48pm UTC: Samczsun performs preliminary audit and finds no issue with patch
- June 11, 2024 at 7:39am UTC: Ultra sound confirms that 0 errors related to the bug have appeared in the ~12 hours since applying the patch on mainnet

Root cause

An incorrectly handled underflow in the calculation of proposer payment based on fee recipient's balance difference could lead to builders inflating their block's value.

We suspect that this bug was introduced because geth moved from bigInt (which prevents underflows) to uint256 when rebasing for Dencun.

The bug is located in the open source Flashbots builder codebase, which is used by MEV-Boost relays to validate blocks submitted by builders.

Impact

This bug caused the ultra sound relay to report an incorrect bid in slot 9265720. We do not have evidence that this impacted the bids reported by other relays or in other slots.

Concretely, the impact of this bug is that the proposer for slot 9265720 may have received a lower value block than they otherwise could have. The precise value lost to the bug cannot be accurately assessed because other block builders adjust their bids based on the top bid reported by MEV-Boost relays, so the existence of an invalid top bid would have affected the sample of alternative bids.

Mitigations

Flashbots developed a patch in collaboration with ultra sound and other MEV-Boost relays.

The patch was audited, unit tested, and run on the ultra sound relay on holesky for ~30 minutes without obvious errors. Flashbots will be upstreaming the audited patch to prevent future issues.

We were able to minimize the impact of this bug thanks to responsible disclosure from members of the block building community. We appreciate their vigilance and judgement in reporting this issue immediately and directly to relay teams. We ask the community to follow their lead in reporting any future vulnerabilities.