

## What

Blind back running is becoming increasingly common in both the L1 and L2 MEV Supply Network. One interpretation of this phenomenon is that the current preference language (bundle types and bespoke APIs) are not sufficient to facilitate this preference in an ironic reversion to PGA-like searching tactics. These blind arb bots are employing clever tactics to avoid detection, as seen in one example where a bot was performing builder balance checks as a signal of being at the top of the block. Catching this is a cat and mouse game, and even more interestingly, showcases an opportunity to express new behavior with a new bundle type.

## How

Since blind back running requires some onchain logic, they typically consider a subset of the pools that a completely offchain arbitrage bot or solver might. Thus if a searcher is able to express which storage slots on a contract they are interested in being inserted behind, builders may be able to more efficiently include them in a block. An example of how this could be expressed is seen below:

```
{ "transactions": [ // ... array of transactions in the bundle ], "blockNumber": "12345678", "minTimestamp": 0,
"maxTimestamp": 100, "revertingTxHashes": [], "storageSlotsChecks": [ { "address": "0xAddressOfContract1",
"storageSlots": [ "0xSlotHash1", "0xSlotHash2" // ... more storage slots ] }, { "address": "0xAddressOfContract2",
"storageSlots": [ "0xSlotHash3" // ... more storage slots if necessary ] } // ... more contracts ] }
```

To integrate, a builder would need to support this bundle type and most likely run this as a separate builder strategy as it may overlap with mechanics of different bundle auctions.

## Discussion

There are a couple of things to consider here. The first is storage slots could be used as a DoS vector or resource attack if left sufficiently unchecked. Some type of sane default must be picked and probably tinkered with. The second is that we once again run in the problem of combinatorial preferences. It's possible an arb of more pools could be more profitable but lose out to another bundle when be considered against only a subset of the first bundles pools. In some ways this idea is a continuation of things with access lists that sort of look like Solana.