🪸

# Primev audit

## Overview

### Links

- Website: https://primev.xyz/
- Blog / Media Center: https://blog.primev.xyz/Primev-Media-Center-68818cfad3d24d60a8b639e9f2021ba1
- Blog / Updates: https://blog.primev.xyz/Primev-Updates-Blog-1096865efd6f80d9ab52cc9697154221
- YouTube: https://www.youtube.com/@Primev_xyz
- GitHub: https://github.com/primev/mev-commit

### Company

**Sources**:

- https://tracxn.com/d/companies/primev/__YP6liccXEwASJYcyPBJrqdS13UmFcEEgyGOjnv2Ucfc
- https://www.crunchbase.com/organization/primev
    - https://www.crunchbase.com/funding_round/t1-protocol-pre-seed--916ce4cf
- https://pitchbook.com/profiles/company/525540-16#overview

**Founded**: 2022

**Headquarters**: San Francisco

**Invested by**:

- Figment Capital
- Andreessen Horowitz
- IOSG Ventures
- LongHash Ventures
- MH Ventures

**Invested into**:

- t1 protocol

## Findings

📢 As a quick summary, our main concerns are:

- Protocol is heavily centralized, all major parts are controlled only by Primev (w/heavy licensing restrictions).
- There were almost no security audits, only on-chain parts were somewhat covered (but not off-chain parts).
- Considering that project exists since mid-2024, current code base looks very hacky & messy (everything is just PoC?).

**Risks:**

- General

  ▼ **[CRITICAL]** Commercially licensed until 1 Oct 2025.

  https://github.com/primev/mev-commit/blob/13eb325f006c58097eab81c378725632166f6beb/LICENSE

  > The Licensor hereby grants you the right to copy, modify, create derivative works, redistribute, and make non-production use of the Licensed Work. The Licensor may make an Additional Use Grant, above, permitting limited production use.

  > If your use of the Licensed Work does not comply with the requirements currently in effect as described in this License, you must purchase a commercial license from the Licensor, its affiliated entities, or authorized resellers, or you must refrain from using the Licensed Work.

  ▼ **[CRITICAL]** Almost no security audits.

  - There was a small one on Cantina (crowd-sourcing) in October 2024.

    announcements ⇒ https://blog.primev.xyz/Primev-Initiates-Security-Audit-with-Spearbit-to-Strengthen-P2P-Network-Infrastructure-1376865efd6f807fbe36f125e59b259e

    conditions & results ⇒ https://cantina.xyz/competitions/4ee8716d-3e0e-4f59-b90d-aa56bf3b484c
  - There are a few notes on general risks, but it is not an audit.

    https://governance.ether.fi/t/primev-symbiotic-risk-analysis/2882
  - Though some audits are planned.

    https://research.lido.fi/t/unlocking-new-validator-yield-with-mev-commit-through-steth/8380#:~:text=We prioritize security%2C rigorously testing mev-commit since the beginning of the year and engaging audit firms and independent security researchers for audits. Security reports will be made publicly available for mainnet.

  ▼ **[HIGH]** Some contracts are upgradable, but there is no DAO or voting to control ownership (as far as we can tell).

  If there is an only person controlling contracts, they may intentionally or unintentionally (due to a hack) upgrade contract to malicious verison.

https://github.com/primev/mev-commit/blob/24df4e8774b2908d824cabfb3eaa017ed64be28d/contracts/contracts/core/Oracle.sol#L18

▼ **[MEDIUM]** Validators ***may*** loose MEV as they will be limited to only use builders, which opted into Primev

https://docs.primev.xyz/v1.1.0/get-started/validators/validator-guide

> As a validator opting into the mev-commit protocol, ensure your mev-boost client only connects to mev-commit opted-in relays to avoid slashing for proposing blocks without delivering commitments.

▼ **[LOW]** Bids are still ?partially? public

1. Bids should only be available to Bidders & Committed Providers

   https://docs.primev.xyz/v1.1.0/concepts/privacy

   > ...commitment and the corresponding bid are only visible to the bidder who made the bid and the provider who made the commitment...

2. Though there are no punishment on commitments

   https://docs.primev.xyz/v1.1.0/concepts/network-overview

   > Only the actors who participated in the block's confirmation are considered for rewards or slashing. This means if Block Builder A and Block Builder B commit to a bid and the target block is built by Block Builder A, the oracle will reward or slash Block Builder A.

3. Though bidders ?may/should/must? somehow choose whom do they send bids

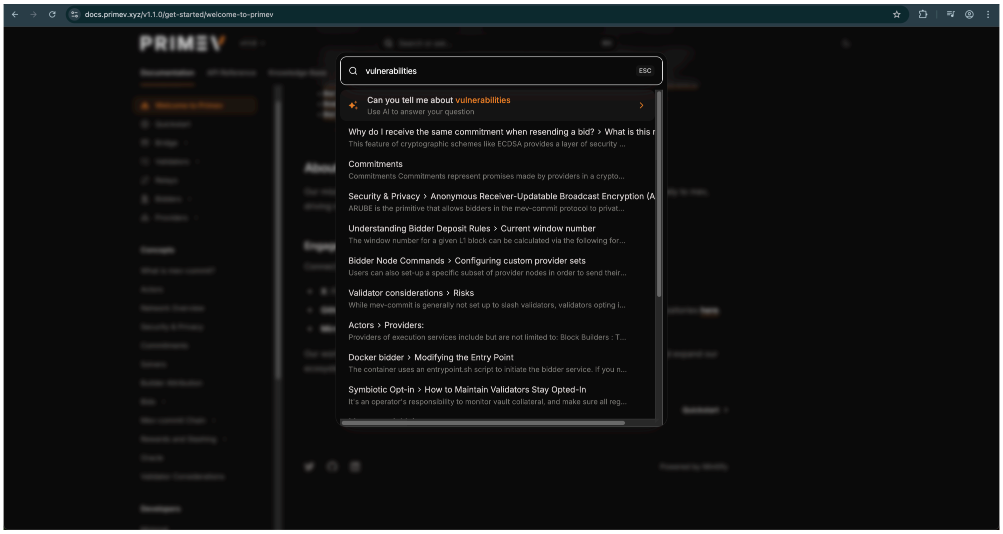
   https://docs.primev.xyz/v1.1.0/concepts/privacy#posting-bids

   > Each bidder can choose a group of providers who will have access to their bids.

▼ **[LOW]** No documented way to report vulnerabilities

No bug bounty program.

No SECURITY.md



vulnerabilities

Can you tell me about vulnerabilities
Use AI to answer your question

Why do I receive the same commitment when resending a bid? › What is this r
This feature of cryptographic schemes like ECDSA provides a layer of security ...

Commitments
Commitments Commitments represent promises made by providers in a crypto...

Security & Privacy › Anonymous Receiver-Updatable Broadcast Encryption (A
ARUBE is the primitive that allows bidders in the mev-commit protocol to privat...

Understanding Bidder Deposit Rules › Current window number
The window number for a given L1 block can be calculated via the following for...

Bidder Node Commands › Configuring custom provider sets
Users can also set-up a specific subset of provider nodes in order to send their...

Validator considerations › Risks
While mev-commit is generally not set up to slash validators, validators opting i...

Actors › Providers:
Providers of execution services include but are not limited to: Block Builders : T...

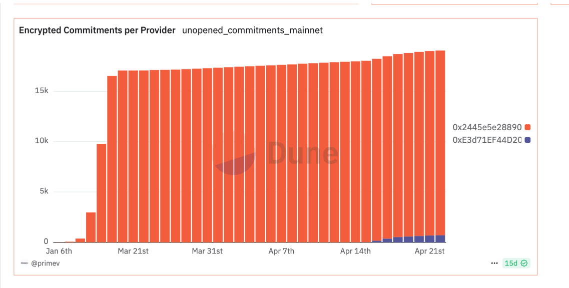
Docker bidder › Modifying the Entry Point
The container uses an entrypoint.sh script to initiate the bidder service. If you n...

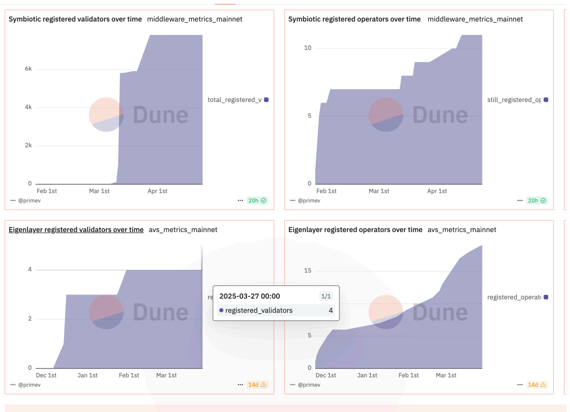
Symbiotic Opt-in › How to Maintain Validators Stay Opted-In
It's an operator's responsibility to monitor vault collateral, and make sure all reg...

▼ **[???]** The Dune stats show suspiciously sparse activity.

- https://dune.com/primev/mev-commit-mainnet-preconf-statistics
- https://dune.com/primev/mev-commit-mainnet
- https://dune.com/primev/mev-commit-mainnet-validator-stats

There are a few spikes, but overall activity is quite flat.





- **GitHub organisation**

▼ **[HIGH]** GHA secrets available to everyone with WRITE access

So called "environments" aren't used in Github Actions to reduce where these secrets are available.

    - https://github.com/primev/mev-commit/blob/13eb325f006c58097eab81c378725632166f6beb/.github/workflows/artifacts.yml#L35
    - https://github.com/primev/mev-commit-oracle/blob/719d2576dc245ab66bf861763165f06fdd675555/.github/workflows/goreleaser.yaml

  ▼ **[HIGH]** GitHub Registry is used, but everyone with Write permissions can publish artifacts.

  https://github.com/primev/mev-commit-geth/blob/6ebd35153400faa54a69bd614d6c5f106052a543/.github/workflows/goreleaser.yaml#L25

- **Oracle**

  ▼ **[HIGH]** Oracle is only operated by Primev

  https://docs.primev.xyz/v1.1.0/concepts/mev-commit-chain/chain-details

  > This is a centralized oracle currently operated by Primev. We're actively looking into decentralizing the oracle role through existing decentralized Oracle protocols and evaluating creating a service where this can be decentralized.

- **Settlement chain**

  ▼ **[HIGH]** There are only two validators at the moment.

  https://docs.primev.xyz/v1.1.0/concepts/mev-commit-chain/differences-between-ethereum-and-mev-commit-chain

  > Mev-commit chain currently operates with only two validator nodes that create blocks in a round-robin fashion. This centralized setup allows for faster consensus and block production. However, it also introduces a higher level of trust in the validators compared to Ethereum's decentralized proof of stake system.

  ▼ **[HIGH]** Chain works in POA

  https://docs.primev.xyz/v1.1.0/concepts/mev-commit-chain/chain-details#:~:text=Mev-commit chain is currently,of block%2C or other services

  This mode creates dependency and requires trust into Primev & validator operators

  ▼ **[MEDIUM]** geth runs in dangerous unlocked mode

  https://docs.primev.xyz/v1.1.0/concepts/mev-commit-chain/chain-details

  > Mev-commit chain is currently built out as an Ethereum sidechain run with go-ethereum's Clique proof-of-authority consensus mechanism

  https://geth.ethereum.org/docs/tools/clef/clique-signing

  > However, using the --unlock flag is generally a highly dangerous thing to do because it is indiscriminate, i.e. if an account is unlocked and an attacker

> obtains access to the RPC api, the attacker can sign anything without supplying a password.

Though:

> Clef provides a way to safely circumvent --unlock while maintaining a enough automation for the network to be useable.

- ▼ **[MEDIUM]** There are FOUR patched geth nodes, which aren’t synced with maintstream
    - https://github.com/primev/mev-commit-geth

      Probably this is ?main? one:
        - https://docs.primev.xyz/v1.1.0/concepts/mev-commit-chain/chain-details#poa-geth-nodes
        - https://docs.primev.xyz/v1.1.0/concepts/what-is-mev-commit#mev-commit-software-components
    - https://github.com/primev/mev-commit-geth-backup
    - https://github.com/primev/go-ethereum-stable-release
    - https://github.com/primev/mev-commit-go-ethereum

- **Relays**
    - ▼ **[CRITICAL]** Relays are only operated by Primev

      https://docs.primev.xyz/v1.1.0/get-started/relays

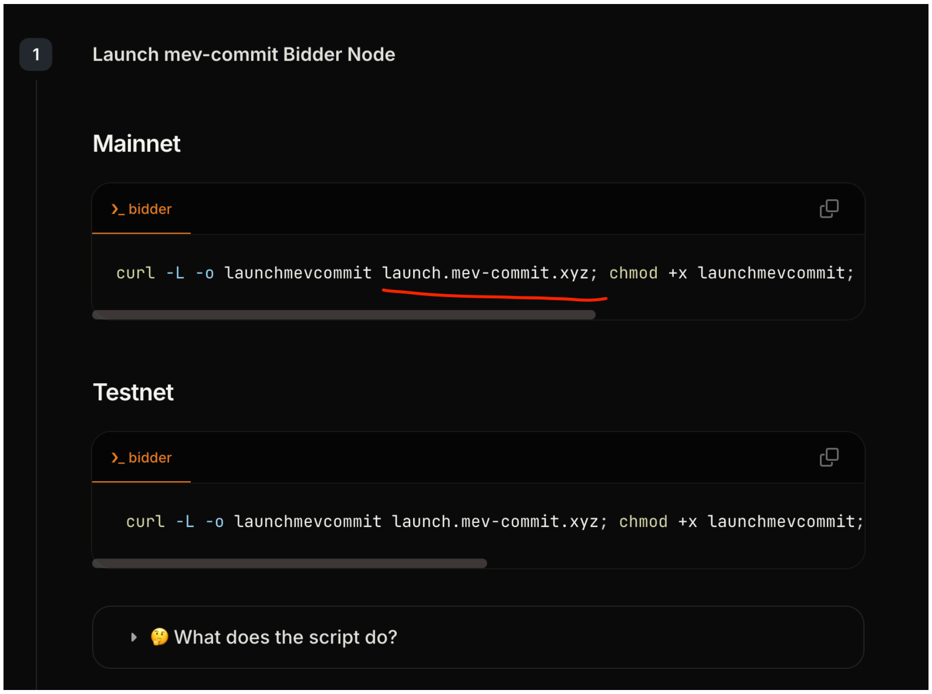
      Relay goes down → all MEV for validators goes down.
    - ▼ **[MEDIUM]** Running patched mev-boost-relay, which also didn’t pass audits

      https://github.com/flashbots/mev-boost-relay/compare/main...primev:mev-commit-relay:main

- **Bidding node**
    - ▼ **[HIGH]** Docs propose to download & execute a script using **HTTP**, not **HTTPS**

      https://docs.primev.xyz/v1.1.0/get-started/quickstart

      ```
      curl -L -o launchmevcommit launch.mev-commit.xyz;
      chmod +x launchmevcommit;
      ./launchmevcommit --node-type bidder
      ```



1 Launch mev-commit Bidder Node

Mainnet

```
curl -L -o launchmevcommit launch.mev-commit.xyz; chmod +x launchmevcommit;
```

Testnet

```
 curl -L -o launchmevcommit launch.mev-commit.xyz; chmod +x launchmevcommit;
```

▸ 🤔 What does the script do?

**Notes:**

▼ How do Validators commit on actually including transaction into block?

https://docs.primev.xyz/v1.1.0/get-started/validators/validator-guide#choose-your-opt-in-method

They do so by participating via EigenLayer/Symbiotic or Native contract:

- Symbiotic

  https://docs.primev.xyz/v1.1.0/get-started/validators/symbiotic

  https://blog.symbiotic.fi/symbiotic-arrives-on-mainnet/

- EigenLayer

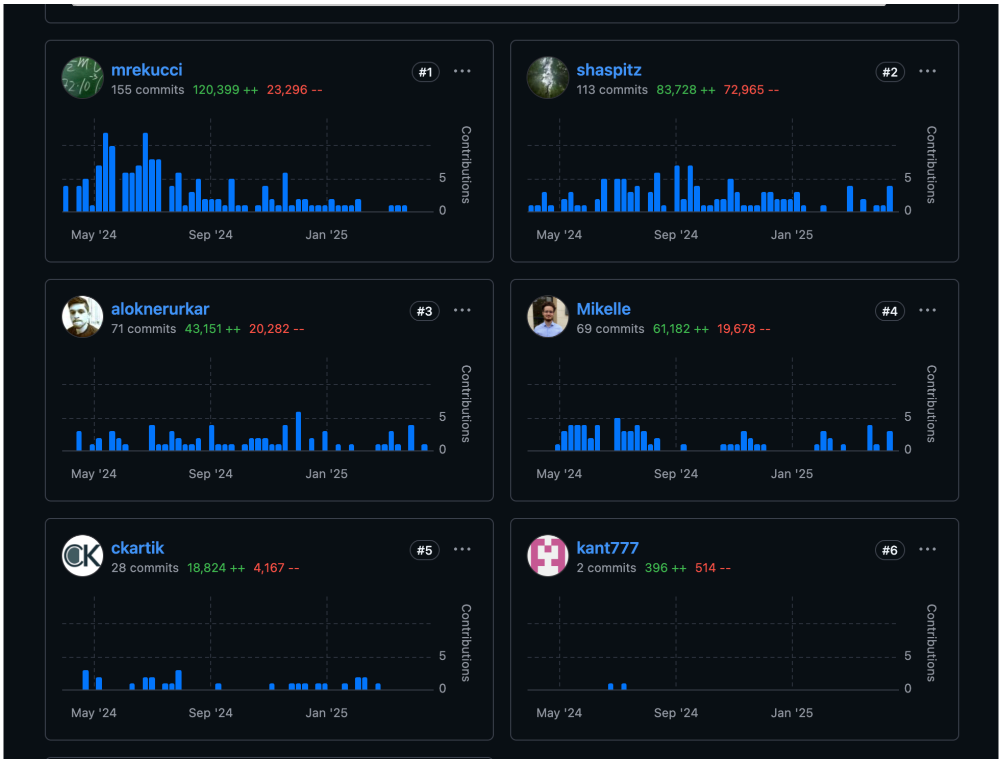
  https://docs.primev.xyz/v1.1.0/get-started/validators/eigenlayer

- Native

  https://docs.primev.xyz/v1.1.0/get-started/validators/vanilla

▼ There are at least 4 active contributors on GitHub

https://github.com/primev/mev-commit/graphs/contributors



mrekucci #1
155 commits 120,399 ++ 23,296 --

shaspitz #2
113 commits 83,728 ++ 72,965 --

aloknerurkar #3
71 commits 43,151 ++ 20,282 --

Mikelle #4
69 commits 61,182 ++ 19,678 --

ckartik #5
28 commits 18,824 ++ 4,167 --

kant777 #6
2 commits 396 ++ 514 --

▼ Works with mev-boost & commit-boost

https://docs.primev.xyz/v1.1.0/get-started/validators/validator-guide#requirements

https://blog.primev.xyz/Mev-commit-Commit-Boost-Seamless-Integration-for-Validators-1ce6865efd6f80e7a4fbfe022bffaa5e?pvs=25

▼ For some reason there are two oracles codebases

Repo 1: https://github.com/primev/mev-commit/tree/13eb325f006c58097eab81c378725632166f6beb/oracle

// docs are pointing to here

Repo 2: https://github.com/primev/mev-commit-oracle/blob/719d2576dc245ab66bf861763165f06fdd675555

- Explorer portal ⇒ https://www.mev-commit.xyz/
- Bridge portal ⇒ https://www.mev-commit.xyz/bridge

# Other

- https://validators.mev-commit.xyz/
- https://www.longhash.vc/post/preconfirmations-credible-promise-of-future-execution
- https://research.lido.fi/t/unlocking-new-validator-yield-with-mev-commit-through-steth/8380
- https://mev-commit-whitepaper.s3.us-east-1.amazonaws.com/mev-commit-whitepaper.pdf