CHAINSECURITY

PUBLIC

Impact Assessment
EIP-8037 and EIP-8038 (Public Version) Ethereum Mainnet

November 18, 2026

Produced for

Ethereum Foundation



Contents

1	Executive Summary	3
2	Introduction and Scope	4
3	Methods and Coverage	5
4	Findings and Mitigations	8
5	Limitations and Use of Report	14

1 Executive Summary

Assessment snapshot: November 18, 2026. Figures derived from the EF findings registry reflect its state on November 13, 2026.

ChainSecurity was engaged by the Ethereum Foundation (EF) to assess the impact of EIP-8037 and EIP-8038, gas-cost changes planned for Glamsterdam, Ethereum's next network upgrade, on existing mainnet systems. We combined a swarm of AI agents reviewing protocols and code with an analysis of a year of mainnet transaction traces.

This is the public version of our assessment. It deliberately omits the names of affected systems, deployments, transactions and any detail that could be used to exploit them. We have submitted most of our detailed EIP-8037 and EIP-8038 findings to the EF findings registry, through which they are shared with affected teams; our Gloas findings are being submitted to the EF separately.

The assessment covers **200 findings: three High, 30 Medium and 167 Low**, with none rated Critical. Our searches are associated with 152 of them, including two High and 22 Medium; the other 48 are associated only with the EF-coordinated investigations. Affected systems include bridges and messaging systems, wallets, exchanges, governance, oracles, staking protocols and supporting off-chain services. The potential consequences include interrupted operations, locked funds and reward-accounting errors.

Fixed gas budgets are the most common source of disruption. These cases break invariant INV-4 of the EF's gas-invariant framework: the assumption that a fixed gas amount remains sufficient for an operation. Affected budgets appear in contract calls and in relayer, keeper and oracle configurations. An insufficient budget may revert the whole transaction or, if the failure is caught, leave an internal action incomplete while the transaction succeeds.

Our searches also identified 22 findings outside the scope of the EF findings registry: 11 concerning Gloas, the consensus-layer part of Glamsterdam, one affecting fee and burn analytics (EIP-7778 with EIP-8037), three caused by other execution-layer EIPs and seven in other off-chain software.

A dedicated search for the proposed INV-9 (plain ETH transfers to new accounts) reviewed 227 systems and produced 39 findings. Mainnet data shows about 105,000 plain ETH transfers per day to new (empty) accounts with less than the 204,600 gas such a transfer will require; unless senders raise their gas limits, these transfers would fail while still paying fees (Section 3.1.3).

We make four principal recommendations:

1. **Give affected teams guidance on how to fix.** Many findings share one pattern: a call fails and execution continues. Intuitive remedies are often incomplete. Without guidance, we expect some fixes to be broken, requiring more rounds of review and delaying the hard fork further. We have proposed such guidance in pull request #4 to the EF's gas-invariant repository (Section 4.4).
2. **Extend the invariant framework.** We have proposed two additional invariants in pull requests #5 and #6 (Section 4.5): INV-9 (21,000 gas no longer guarantees a plain ETH transfer to a new account) and INV-10 (`eth_estimateGas` may return too little gas when a contract tolerates a failed inner call). We have submitted a first set of INV-9 findings to the EF findings registry and have additional findings for both invariants.
3. **Consider the effects of other EIPs and Gloas.** Our searches found compatibility issues caused by Gloas and by other execution-layer EIPs (Section 4.6). The EF should assess whether the effects of other EIPs, such as EIP-7708 or EIP-8246, and of Gloas have been sufficiently considered, both on their own and in combination with EIP-8037 and EIP-8038, and establish a coordinated process for consensus-layer compatibility issues.
4. **Allow sufficient time before activation.** 24 findings require contract upgrades or migration, and all three High-rated findings require action before activation (Section 4.3). The schedule should allow affected teams to validate and deploy remedies, particularly where funds are at risk or governance coordination is required.

2 Introduction and Scope

Glamsterdam is Ethereum's next network upgrade. It combines Amsterdam, the execution-layer changes, with Gloas, the consensus-layer changes. In this report, Osaka denotes the current execution-layer rules and Amsterdam the proposed ones. EIP-8037 changes state-creation gas costs and accounting, including a separate state-gas reservoir: gas set aside for creating state, which can be returned when execution reverts. EIP-8038 reprices state access. Together, they can invalidate existing assumptions about gas budgets, failure handling and gas accounting. This assessment examines the consequences for Ethereum mainnet systems and the actions needed to preserve their operation.

The work builds on the investigation accompanying ChainSecurity's blog post Glamsterdam Hard Fork: When Spent Gas Comes Back. It combines additional searches with review and validation of existing findings. We assess impact, recovery options and mitigation feasibility, particularly where action is needed before activation.

The EF coordinates a findings registry that provides a complementary view of affected mainnet systems. It brings together AI-assisted searches and contributions from other teams. The EF-coordinated investigations combine scans for gas-sensitive code with targeted protocol review. They use source analysis, transaction history and replays under Osaka and Amsterdam rules to assess failures, operator gas settings and recovery options. The registry is not public. This report includes aggregate statistics derived from it, but no individual records.

We organize gas-related findings using the EF's gas-invariant framework. It extends the assumptions described in our blog post and provides a common vocabulary for explaining how they can break. Our proposed additions to the framework are discussed in Section 4.4 and Section 4.5.

The primary scope is Ethereum mainnet contracts and off-chain services that submit, relay or account for mainnet operations. L2-only findings are excluded. Cross-chain configurations are included where they control execution on mainnet. We discuss findings outside the registry's scope, including those concerning Gloas, separately.

The engagement covers identification, impact analysis and assessment of mitigation options. Fix implementation and review, reassessments, protocol outreach and ongoing support are outside scope. The conclusions apply to the specification revisions and evidence assessed. Coverage is not exhaustive, and the assessment does not certify the security of the systems reviewed.

Disclosure. Many findings concern systems that have not yet been fixed. Findings in the EF findings registry are shared with affected teams through the EF's coordinated disclosure process. This public version therefore reports aggregate results, mechanism classes and recommendations only. It does not identify affected projects, contracts, services, client implementations or transactions, and does not describe how individual failures can be triggered.

3 Methods and Coverage

3.1 Work performed and evidence

3.1.1 Agent-swarm search

We used a coordinated swarm of AI agents from the Claude and GPT model families to search for affected contracts and off-chain components. Agents investigated different protocol categories in parallel, shared leads and tools, and assigned proposed findings to separate reviewers.

Discovery combined protocol and deployment lists, source and bytecode scans, on-chain event and transaction censuses, and review of public operator software. Searches targeted fixed gas budgets, tolerated call failures, gas-based accounting and estimation assumptions. Findings prompted searches for related implementations and integrations; a pattern match alone was not a finding.

Candidates were investigated using a simulator built on Ethereum's execution specifications. It executed the same scenario under the baseline Osaka rules and the proposed Amsterdam rules, with pinned state and gas parameters. Historical transaction replays were the primary evidence, supplemented by constructed scenarios for rare paths and adversarial inputs. Historical replays required an exact match to the baseline receipt's gas usage. Investigations varied gas budgets and relevant state, examined both internal calls and transaction outcomes, and checked recovery paths. They also distinguished fixed sender limits from per-transaction estimates and their buffers.

Separate agents challenged proposed findings by checking whether failures also occurred under Osaka and whether recovery remained possible. The review protocol required a test designed to disprove each finding. Proposals rated Medium or higher also required adversarial refutation and two severity assessments blinded to the author's proposed rating. Not every check was completed for every finding. After review, we excluded five swarm-confirmed findings from the assessment population: one pre-existing issue, two that recover within the block gas limit, and two that only matter if L2 chains adopt the changes.

3.1.2 Trace-based search

We analyzed approximately one year of Ethereum transaction history and internal call traces to identify contracts and senders involved in real mainnet activity. This included contracts reached indirectly through other contracts. Similar implementations were grouped for review, while call relationships helped expand individual leads into related deployments and shared infrastructure.

Screening combined code patterns with historical gas usage and state changes to identify operations sensitive to the proposed gas changes. Searches targeted fixed gas budgets, tolerated call failures and gas-based accounting, with candidates prioritized by activity and potential exposure. Following the calls also helped identify which contract or operator controlled a gas allowance.

AI reviewer agents investigated selected cases using deployed code and real transactions replayed under Osaka and Amsterdam rules, with the baseline checked against historical receipts. They examined the cause and consequence of each failure, whether the operation could be retried, and who would need to act to restore it. Separate AI challenger agents attempted to refute the conclusions, including by checking alternative explanations and recovery paths.

Historical activity broadened discovery but did not establish that every contract in the collected data was assessed. Coverage remained limited by the observation window, filtering decisions and the cases reviewed. Dormant paths and off-chain gas settings required additional targeted investigation.

3.1.3 Targeted search for INV-9

Between November 16 and 18, 2026, we ran a dedicated search for off-chain code affected by the proposed INV-9. Candidates were reproduced on the execution specifications under Amsterdam rules, using re-

implementations of the pinned project code and, for some findings, the projects' own code against a local node. Pairs of blind raters assessed severity, and a separate reviewer without prior context re-examined every finding. The search covered 227 systems and produced 41 candidate findings. After review, two were withdrawn, leaving 39.

A census of recent mainnet activity found approximately 105,000 plain ETH transfers per day to new (empty) accounts with a gas limit below the 204,600 gas such a transfer requires under the proposed rules. Without changes, these transfers would be included with a failed status and move no ETH.

3.1.4 Contributions

We opened 90 pull requests to the EF findings registry. Of these, 76 each add a new EIP-8037 or EIP-8038 record, and one updates attributions and confirmations on 76 existing records. These contributions document affected deployments, evidence and proposed remedies. Some further confirmations and extensions of existing records have not yet been submitted. The remaining 13 pull requests concern INV-9: 12 add a first set of INV-9 records, and one proposes that INV-9 be added to the registry's own invariant list (separate from pull request #5 to the framework). In addition, we proposed fix guidance and two additional invariants for the EF's gas-invariant framework as open pull requests (Section 4.4 and Section 4.5).

3.2 Coverage achieved

Approach	Findings
Agent-swarm search	146
Trace-based search	58

The approaches overlap on 52 findings, giving **152 unique findings**, summarized in Section 4.

The swarm's topic lists contained 17,663 addresses, supplemented by other discovery methods. The trace-based search collected 1,319,692 distinct runtime bytecodes from observed mainnet activity.

The EF-coordinated investigations identified **48 findings absent from our results**, including deployments we missed or had assessed as unaffected. These gaps show that our coverage was incomplete and suggest further affected systems remain unidentified. Together with the EF findings, the combined assessment covers **200 findings** (197 confirmed and three candidates awaiting full confirmation).

Our searches also identified **22 findings outside the registry's scope**, excluded from the 200-finding population and discussed in aggregate in Section 4.6.

3.3 Limitations and remaining uncertainties

- **Coverage and selection.** Discovery depended on protocol inventories, screening rules and historical activity. Rare or dormant paths, recent deployments and differences between similar implementations may escape review.
- **Source availability.** Source-based investigation relied on verified code available through services such as Sourcify and Etherscan. Missing or incomplete sources limited analysis depth and identification of affected implementations. Unverified contracts could still be investigated through bytecode analysis and transaction replays.
- **Evidence and review.** Evidence ranges from historical replays to constructed scenarios and source analysis. Separate reviewer and challenger agents reduce errors, but can share assumptions or overlook problems in common tooling.

- **Off-chain visibility.** Off-chain review relied on available source code and observable transactions. Private services, internal integrations and deployed software versions were not fully visible. Reproducing a failure in reviewed code does not establish how widely that code is deployed or how a production service handles it.
- **Operator behavior.** Observed transactions reveal gas settings in use, but do not always expose private configuration or estimation policies. The practical impact can depend on whether an operator adjusts its gas allowance, retries an operation or changes its software.
- **Exposure and mitigation.** A demonstrated failure does not establish the total value exposed across every deployment. Proposed remedies also require case-specific validation, particularly for existing positions, consumed authorizations and in-flight messages.

4 Findings and Mitigations

4.1 Consolidated overview

The assessment brings together **200 EIP-8037 and EIP-8038 findings**. Of these, **124 findings are associated with the EF-coordinated investigations** and **152 with our combined agent-swarm and trace-based searches**. The comparison identifies **76 findings covered by both, 48 covered only by the EF-coordinated investigations, and 76 covered only by our searches**. These are attribution counts, not claims of independent discovery.

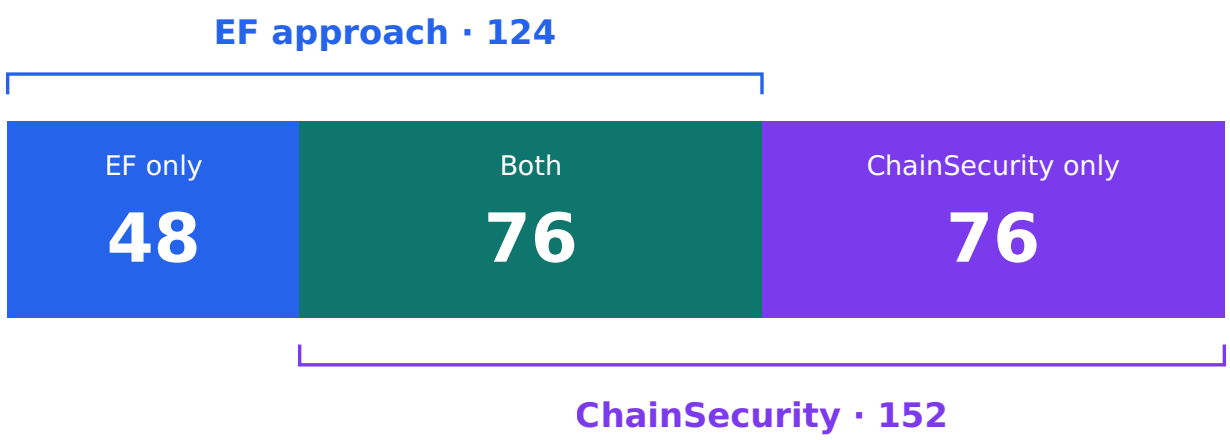


Figure 1. Overlap between the EF-coordinated investigations and the combined ChainSecurity searches.

Our combined swarm and trace-based results cover **152 unique findings: two High, 22 Medium and 128 Low**, with none rated Critical. Each finding counts once and may cover multiple deployments. The full 200-finding population includes **three High, 30 Medium and 167 Low** findings.

Ratings follow the EF's mitigation-focused rubric, considering who must act, whether recovery can be relied on, and the consequences without action. Simple operator changes or retries generally rate **Low**; upgrades, software releases or coordination generally rate **Medium**. Loss or locking without dependable mitigation can rate **High**, with permissionless theft or destruction potentially **Critical**. Limited exposure and dormant paths can lower ratings. **Info** marks latent issues without current impact, and ratings marked *conditional* apply only if stated preconditions hold; both appear only in Section 4.6.

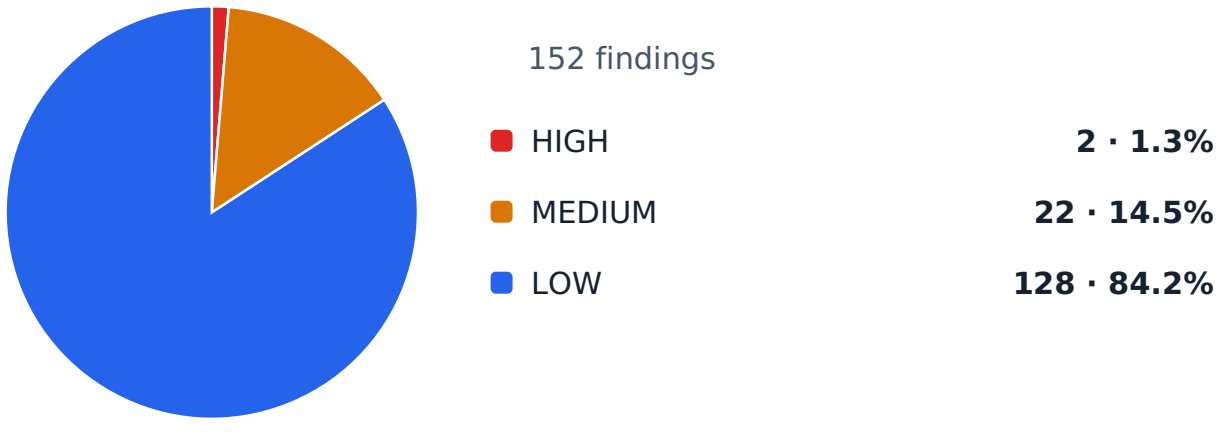


Figure 2. Severity of the 152 combined agent-swarm and trace-based findings. Each finding counts once; the other figures use the full 200-finding population.

Fixed gas budgets are the dominant source of disruption: 182 findings (91.0%) affect INV-4, the assumption that a fixed gas amount remains sufficient. Other findings concern gas reserves and stipends, retained gas, gas returned after reverts, immediate gas refills when temporary storage is cleared, and gas-based accounting. A finding can affect several invariants, giving 248 invariant associations across the 200 findings.

Invariant: change and consequence	HIGH	MEDIUM	LOW	Total
INV-4 · Fixed gas budgets Previously sufficient budgets become too small; calls or operations fail.	2	28	152	**182**
INV-7 · Gas reserves and stipends Existing reserves and stipends no longer cover the intended work.	1	3	16	**20**
INV-6 · Retained gas after a call The retained 1/64 can finish the caller despite an inner call failing.	0	3	15	**18**
INV-3 · Gas returned on revert Reverts return state gas, defeating gas-based failure checks.	1	0	13	**14**
INV-2 · Immediate storage refills Clearing temporary storage returns gas immediately, funding later work.	0	1	7	**8**
INV-5 · Gas-based accounting Gas readings can rise or omit reservoir costs, distorting fees and refunds.	0	1	5	**6**
INV-1 · Failure-path state writes Reservoir gas can fund state writes that permanently record a failed call.	0	0	0	**0**
INV-8 · EIP-7702 authorizations A failed delegation can leave an included authorization reusable.	0	0	0	**0**

Figure 3. How the proposed changes affect each of the eight EF invariants, and severity of associated findings. Ratings apply to findings, not invariants. Findings can appear in several rows; zero means none identified in this assessment.

4.2 Potential consequences

Without mitigation, the findings can lead to **liveness disruption (interrupted operations) in 148 cases, locked funds in 33, loss of funds in 13, other consequences in four and griefing in two**. Each finding is counted once under its worst consequence. The counts indicate the distribution of potential impacts, not their likelihood or financial scale. Because ratings depend on who must act and whether a dependable mitigation exists, most loss-of-funds findings are rated Low or Medium.

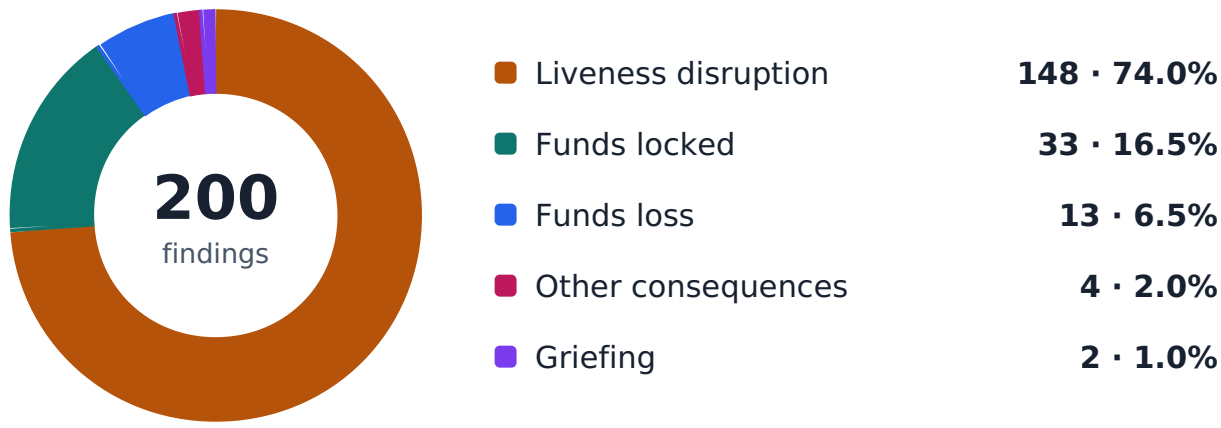


Figure 4. Recorded worst consequence per finding.

4.3 Mitigation feasibility and limitations

Configuration changes are the primary remedy for **141 of 200 findings**. Another **24 require contract upgrades or migration**, where permissions, coordination and existing positions can constrain recovery. All three High-rated findings require action before activation by the responsible parties. Figure 5 shows the full breakdown. Each finding counts once, although its remedy may involve several actions.

Configuration and operational changes are generally the simplest remedies. They may involve increasing gas allowances, adjusting how operations are submitted, or temporarily pausing affected operations. The change must cover the full operation, including recovery after a failure.

Contract upgrades and migrations require more coordination. Protocol owners, governance bodies, external service providers or operators on other chains may need to act. Where contracts cannot be upgraded, users may need to move their positions.

Mitigation type	Severity			
	HIGH	MEDIUM	LOW	Total
Off-chain configuration	0	19	85	104
Operational change	1	2	21	24
On-chain configuration	0	4	33	37
Change on another chain	0	1	8	9
Contract upgrade	1	3	10	14
Migration / redeployment	1	1	8	10
Unknown / not classified	0	0	2	2

Figure 5. Primary mitigation category by severity.

Remedies should be validated against both new operations and the recovery of failed or pending operations before deployment.

4.4 Guidance on fixing

Many findings share a single pattern: an external call fails and the calling code continues. Under the new gas rules, whoever chooses a transaction's gas limit can sometimes cause such a call to fail while the outer transaction succeeds. This pattern underlies several invariants in the framework (INV-1, INV-2, INV-3, INV-6 and INV-7).

Assessing mitigation options showed that intuitive remedies are often incomplete. Raising a gas allowance may not cover recovery of failed or pending operations. A gas check before a call can be miscalibrated under the new accounting. A replacement contract can protect future activity while leaving existing deployments exposed. Making a failed operation retryable requires that the failure is not recorded as final.

We advise the EF to give affected teams guidance on how to fix. Otherwise, some fixes are likely to be broken, and the resulting rounds of review would delay the hard fork further. We have made a proposal in pull request #4 to the EF's gas-invariant repository: a short decision guide for handling call failures, with five rules:

- **Propagate the failure**, so that the whole operation reverts.
- **Check first, then call**: check preconditions such as balances and approvals, so that failures are anticipated rather than tolerated.
- **Make the failure retryable**, so that it does not consume a message, claim or authorization.
- **Check the gas before the call**, accounting for the new gas rules.
- **Limit the gas the callee can use**, where a bounded callee is intended.

Each rule includes examples and pitfalls.

4.5 Additional invariants

During our searches, we identified affected systems relying on two assumptions not yet covered by the framework's eight invariants. We have proposed both as additional invariants in pull requests #5 and #6:

- **INV-9: 21,000 gas no longer guarantees a plain ETH transfer to a new account.** With EIP-2780 and EIP-8037, a value transfer to a new account requires more than the 21,000 gas commonly assumed. A transaction with a lower limit can still be included with a failed status, paying the fee and consuming the nonce without moving ETH. Systems that hard-code a 21,000 gas limit, rely on pre-signed transactions or process incoming ETH transfers need review.
- **INV-10: `eth_estimateGas` may return too little gas when a contract tolerates a failed inner call.** Gas estimation searches for a limit at which a transaction does not revert. Where a contract tolerates a failed inner call, the estimate can fall in a range where the transaction succeeds while the inner work fails. The new gas rules can open such ranges in code that previously estimated correctly.

Our targeted INV-9 search (Section 3.1.3) produced 39 findings. We have submitted a first set of 12 to the EF findings registry as open pull requests. Their details and severity distribution are withheld from this public version. **We have additional findings for both invariants.** Neither the 39 INV-9 findings nor the additional findings are included in the 200-finding population or the figures. Some overlap with findings discussed in this report, including several of the 22 in Section 4.6. We recommend that the EF incorporate both invariants into its framework and coordinated searches.

4.6 Findings outside the registry's scope

Glamsterdam includes many changes besides EIP-8037 and EIP-8038. The gas-invariant framework and the findings registry focus on those two EIPs, while some of the issues we identified arise from other changes, from combinations of changes, or concern software outside the registry's scope. **The EF should assess whether the effects of other EIPs, such as EIP-7708 or EIP-8246, and of Gloas have been sufficiently considered, on their own and in combination with EIP-8037 and EIP-8038.** For example, EIP-7708 makes ETH transfers emit logs that off-chain consumers of transfer events may not expect, and EIP-8246 stops `SELFDESTRUCT` from burning ETH, so balances that were previously destroyed now remain in balance-only accounts. We have not reported findings caused by EIP-8246; it illustrates a change whose effects on existing systems warrant assessment.

Our searches identified **22 findings outside the registry's scope**. They are excluded from the 200-finding population and the figures above. Severities and per-finding details beyond those stated for Gloas are withheld from this public version. Additional Gloas cases from trace-based search (Section 4.6.1) are not included.

Area	Findings	Principal effects
Gloas (EIP-7688, EIP-7732, EIP-8061)	11	Blocked staking operations, stale accounting, interrupted services, outdated checkpoint-limit assumptions
Fee and burn analytics (EIP-7778 with EIP-8037)	1	Incorrect fee and burn figures
Other execution-layer EIPs (EIP-7708; EIP-7976, alone and with EIP-2780)	3	Misclassified transfer data, gas-accounting differences and interrupted transaction submission
Other off-chain software	7	Assumptions about plain-transfer gas, gas-limit handling and estimation

4.6.1 Gloas compatibility

Gloas, the consensus-layer part of Glamsterdam, introduces compatibility issues separate from the gas changes. The 11 findings from our agent-swarm search (five Medium, five Low and one Info) fall into three mechanism classes. Trace-based search covered the same on-chain verifiers and a few additional ones, rating them Low or Info. It also identified two further conditional Low-rated cases, separate from these verifiers, whose details are withheld from this public version. None of these additional cases are counted in the 11. None of the 11 findings demonstrated a permanent loss of funds.

- **Beacon-state proofs.** EIP-7688 changes how beacon-state data is arranged in Merkle trees. On-chain verifiers with fixed proof layouts reject valid post-fork proofs, which can block staking operations and leave accounting stale. EIP-7732 also changes data that some proofs rely on, so these need a new data source rather than new proof paths alone. Proof generators and verifiers must be updated together, and tests must cover operations that span the fork or the verifier upgrade.
- **Execution data location.** Under EIP-7732, execution payloads are carried separately from the beacon block body. Off-chain services that read the old location can stop processing blocks. Builder payments also introduce withdrawal entries that do not refer to validators, which services must distinguish.
- **Checkpoint trust limits.** EIP-8061 increases exit and consolidation churn (the rate at which validators can leave or merge) and so shortens the weak-subjectivity period. Implementations that rely on weak-subjectivity checkpoints should derive their limits from the active fork and validator state.

These findings are outside the scope of the EF findings registry and are being submitted to the EF separately. We recommend a coordinated process for consensus-layer compatibility issues comparable to the one established for EIP-8037 and EIP-8038.

4.6.2 Fee and burn analytics

EIP-7778 excludes transaction refunds from block gas accounting, while EIP-8037 meters execution and state gas separately. Block-header gas usage can therefore differ from the total settled in transaction receipts. Off-chain calculations that assume equality between header and receipt gas usage can produce incorrect figures. Calculations must use receipt-based accounting appropriate to the quantity being calculated.

4.6.3 Other execution-layer EIPs and off-chain software

Three findings arise from other execution-layer EIPs: EIP-7708, and the calldata-cost changes in EIP-7976 on their own and in combination with EIP-2780. Seven further findings concern off-chain software outside the registry's scope, including assumptions about the gas needed for plain transfers, gas-limit handling and gas estimation. Several relate to the proposed INV-9 and INV-10, and some have since been incorporated into our INV-9 submissions (Section 4.5).

5 Limitations and Use of Report

Impact assessments cannot identify every affected system; an assessment that does not identify a system as affected is not a guarantee that the system is unaffected. However, impact assessments enable the discovery of assumptions that were overlooked during development and of areas where action is needed before a network upgrade activates. Some issues affect a system as a whole, while others arise only in particular code paths, configurations or operating conditions. Within the agreed time frame, ChainSecurity has performed an assessment aimed at identifying as many affected systems and mechanisms as possible.

The focus of our assessment was limited to the scope defined in the engagement letter and described in this report. Our conclusions are based on the specification revisions, deployed code, transaction history, information from the EF-coordinated findings registry and other evidence available at the time of the assessment. Registry data contributed by other teams was reviewed but not independently reproduced in every case. Specifications may still change before activation, and deployed systems, their configurations and their usage may change at any time; such changes can alter the conclusions of this report. We draw attention to the fact that due to inherent limitations in any software, analysis tooling and assessment process, an inherent risk exists that even major failures or malfunctions remain undetected. We did not systematically assess the correctness of client implementations or third-party infrastructure. Where our analysis relied on them, for example for transaction replays, we assumed they behave correctly, which adds further inherent risks. This report does not certify or endorse the security of any system mentioned or described in it, and readers should not rely on it as such.