Filecoin Actors Audit

Date	September 2020
Lead Auditor	Alexander Wade
Co-auditors	Daniel Luca

1 Executive Summary

From June to September 2020, Consensys Diligence engaged with Protocol Labs to assess the security of Filecoin's builtin Actors: a collection of executable code that implements the core business logic of Filecoin's blockchain-based storage network.

We began in June with a 2-week preliminary phase, which we used to get familiar with Filecoin: reading code and documentation, asking questions, and collecting notes to prepare for an intensive security assessment. During this phase, we identified two important foci which carried over into upcoming work:

- **Coordination**: Filecoin's Actors were under heavy development for the duration of this assessment, as the project was preparing for several milestones: an initial testnet release, an incentivized testnet release (the Filecoin Space Race), and an eventual mainnet release. We needed to keep up with the pace of changes, and

we needed to ensure the Actors devs were aware of anything that needed their attention. To these ends, we created and shared Actors' Master Tracking and continued to maintain it for the duration of the engagement.

- **Documentation**: To review a complex codebase like Filecoin's Actors, we needed to know how it was supposed to work. After finding that existing documentation was either out-of-date or nonexistent, we began documenting system behavior. Documentation was an ongoing effort during this engagement and was tracked in Master Tracking: Documentation.

We continued with an intensive 9-week assessment of Actors code, focused primarily on its role in implementing the core business logic of the Filecoin storage network. A detailed description of what was and was not reviewed can be found in Scope.

Our responsibilities were as follows:

- **Analyze code**: We performed a manual review of Actors code, primarily to identify flaws in the implementation of its business logic. Our secondary focus was to suggest improvements or simplifications that increased the code's robustness, and to raise discussions about the purpose of various implementation details. Any outputs from these foci were immediately communicated via GitHub issue. Actors devs reviewed these outputs, assigning each a label which determined its priority relative to mainnet launch. A complete record of these findings can be found in Findings.
- **Review incoming changes**: As the Actors code was under heavy development, we attempted to review as many incoming changes as possible. This included a cursory review of most commits and an in-depth review of some pull requests. Where applicable, we also updated our documentation to reflect these changes.
- **Maintain the Master Tracking Doc**: We updated Actors' Master Tracking daily to highlight anything that needed attention: open issues, engagement schedule, changes to documentation, and more.
- **Attend bi-weekly calls with Actors devs**: We attended bi-weekly calls with Actors devs to understand what was being worked on, ask questions, and communicate any blockers.

Our work concluded on September 11, 2020, with the creation of this report.

Update (Oct 16, 2020): We engaged with Protocol Labs for an additional two weeks, from Oct 5 to Oct 16, 2020. Our objectives for this period were to review changes

made since September 11, update the status of any previously-filed issues, and investigate additional components of Actors code.

2 Scope

This assessment's primary focus was to review that code in the filecoin-project/specs-actors repository most pertinent to the function of Filecoin's builtin Actors. The review was centered on Go files (`*.go`) within the `/actors` directory.

Of these files, this assessment was **not** concerned with:

- Any non-Go files, such as `reward_calc.py`
- Test files (`*_test.go`) outside the `/actors/builtin` directory
- CBOR-Gen (`cbor_gen.go`) files anywhere in the specs-actors repository

Of these files, this assessment was **less** concerned with:

- Files outside the `/actors/builtin` directory

Additionally, the following was **out of scope**:

- Implementation of and usage of dependencies, including (but not limited to):
  - `filecoin-project/go-address`
  - `filecoin-project/go-amt-ipld`
  - `ipfs/go-hamt-ipld`
  - `filecoin-project/go-bitfield`
  - `ipfs/go-cid`
  - `ipfs/go-ipld-cbor`
  - `minio/blake2b-simd`
  - `minio/sha256-simd`

    - `multiformats/go-multihash`
    - `whyrusleeping/cbor-gen`
- The Lotus client, including (but not limited to):
    - Implementation of runtime interface exposed to builtin actors
    - Storage Power Consensus implementation
    - Block/Epoch/Tipset processing
    - Message/signature verification
    - Networking components
    - PoRep / PoSt
    - Filecoin Gas mechanism
- Correctness of cryptoeconomic incentives and supporting implementation:
    - Parameters used for monetary policy, incentives, penalties, power accounting
    - Block/Epoch reward calculation and smoothing

3 Findings

As we uncovered vulnerabilities and came up with potential improvements and simplifications, we opened issues in the filecoin-project/specs-actors repository. After reviewing our findings, Actors devs assigned each a relative priority:

- **Priority 1 (P1)**: Required for mainnet. Reserved for vulnerabilities or otherwise significant changes to implementation.
- **Priority 2 (P2)**: Beneficial for network launch. Reserved for minor vulnerabilities or changes that would improve code quality or robustness but do not represent a pressing need.
- **Priority 3 (P3)**: Not urgent or important.

All issues we opened are listed below, grouped by this relative priority. Note that the status of these issues represents the status at the time of report creation. Up-to-date status and information on all mentioned issues can be found by following the provided links.

Update (Oct 16, 2020): Our followup engagement took place during Filecoin's transition from testnet to mainnet. Because the priority levels described above only make sense in a pre-mainnet context, any issues opened during this period are

included under Followup Work. Information on further work performed on these issues can be found by following the provided links.

3.1 Priority 1

specs-actors Issue	Status	Title
#587	Closed in #588	Multisig: Comparison between different Address types.
#602	Closed in #718	Market: withdraw allows anyone to trigger withdrawals
#606	Closed in #620	Market: Deal States update not persisted in CronTick
#612	Closed in #646	PaymentChannel: vouchers can be replayed across channels
#643	Closed in #644	Market: CronTick doesn't persist Proposal deletion from st.PendingProposals
#660	Closed in #778	PaymentChannel: Reject equal-nonce voucher submissions
#692	Closed in #791	StoragePower: OnConsensusFault incorrectly zeroes out miner power
#733	Closed in #790	PayCh.Collect: Clarify whether Collect implies the channel should be terminated
#753	Closed in #789	Miner: Incorrect bounds on SubmitWindowedPoSt params.Deadline
#755	Closed in #760	Market.PublishStorageDeals: Validate each deal's PieceSize
#765	Closed in #775	Market State.dealGetPaymentRemaining: Bad assertion allows for panic
#766	Closed in #775	Market.CronTick: Branching statements imply potential abort

specs-actors Issue	Status	Title
#767	Closed in #775	Market.CronTick: Remove RequireSuccess for interaction with VerifiedRegistry and BurntFundsActor
#797	Closed in #890	Reward.AwardBlockReward: Miner may be able to cause abort during block processing
#909	Closed in #1073	Power: Process batch proof verifications before deferred cron events
#982	Closed in #1050	Miner: Remove ChangeWorkerAddress reliance on cron
#1008	Closed in #1089	Miner: Verify that duplicate submissions of consensus faults are not processed
#1056	Closed in #1092	Power: Explicitly delete miner claim on failing cron callback
#1100	Closed in #1129	Miner: Declaring a replaced CC sector faulty can result in a sector existing in an expiration queue twice

3.2 Priority 2

specs-actors Issue	Status	Title	Comment
#751	Open	Simplify cron queue handling in power and market actors	From @anorth: “We’ve determined that this isn’t high importance for network launch. It’s a good simplification, but not trivial to ensure correctness. The current implementation has the benefit of having been run in testnets for quite some time.”

specs-actors Issue	Status	Title	Comment
#931	Open	Miner: ExtendSectorExpiration doesn't correctly check AddressedPartitionsMax	From @anorth: "We've determined this isn't high importance for network launch. The check as implemented is more conservative than we would like to allow. This is inconvenient for miners, but I don't think poses a risk."
#979	Open	Where applicable, enforce uniqueness when handling slices	From @anorth: "We've determined this isn't a critical change needed for network launch. As we approach that point, adding more constraints on the node implementations adds risk there. It is still something I'd like to follow up with later just to reduce degrees of freedom."
#981	Open	Miner/Market: Clean up deal weight calculation during precommit	From @anorth: "We've determined that this isn't high importance for network launch. It's a nice clean-up that we'll implement in a discretionary upgrade later."
#1006	Open	Miner: Cleanup accounting methods	From @anorth: "We have determined that these are not critical changes for network launch. They represent a solid clean-up and simplification, but as we approach mainnet, changing introduces new risk. I hope to land these in a subsequent discretionary upgrade."

specs-actors Issue	Status	Title	Comment
#1020	Open	Miner: VerifyPledgeRequirementsAndRepayDebts should return balance available after paying debt	From @anorth: “We have determined that these are not critical changes for network launch. They represent a solid clean-up and simplification, but as we approach mainnet, changing introduces new risk. I hope to land these in a subsequent discretionary upgrade.”
#1060	Open	Add log messages for significant events	From @anorth: “We’ve determined this is not critical for network launch. It will greatly aid troubleshooting, but does not itself resolve or reduce any specific risk.”
#608	Closed in #647	Market: ComputeDataCommitment loads identical AMT for each passed-in DealID	
#609	Closed in #647	Market: CronTick loads identical AMT for each processed deal	

specs-actors Issue	Status	Title	Comment
#697	Closed in #862	Check if provided param.Penalty is greater or equal to zero	
#722	Closed in #758	Verifreg: Disallow RootKey from assuming other roles	
#724	Closed in #758	Verifreg.AddVerifier: params.Allowance should be greater than or equal to MinVerifiedDealSize	
#725	Closed in #758	Verifreg.AddVerifiedClient: Existing Verifiers should not be able to become VerifiedClients	

specs-actors Issue	Status	Title	Comment
#726	Closed in #758	Verifreg.AddVerifiedClient: Misleading parameter “MinVerifiedDealSize”	
#728	Closed in #758	Verifreg.UseBytes: Consider never deleting a VerifiedClient entry	
#729	Closed in #939	Verifreg: All methods should resolve addresses to ID-addresses before interacting with state	
#730	Closed in #912	Multisig, verifreg, paych actors create accounts that don’t already exist	

specs-actors Issue	Status	Title	Comment
#752	Closed in #808	Miner: Duplicate invocations of notifyPledgeChanged in processEarlyTerminations	
#771	Closed in #775	Market.AddBalance creates balance table entries when provided 0 value	
#795	Closed in #919	Miner: Sector activation at ProveCommit may allow proving expired/expiring sectors, leading to panic during cron tick	
#798	Closed in #820	Miner.PreCommitSector: Conflicting MaxSectorNumber checks	

specs-actors Issue	Status	Title	Comment
#802	Closed in #902	Miner.ChangeWorkerAddress: handle if pending change already exists	
#806	Closed in #1050	Limit Miner.ChangeWorkerAddress to avoid cron event queue load	
#945	Closed in #984	Miner: Limit max number of partitions per deadline	
#977	Closed in #998	Miner: Limit size of ControlAddresses slice	
#983	Closed in #1009	Miner: reduce operations performed in handleProvingDeadline where possible	

specs-actors Issue	Status	Title	Comment
#1003	Closed in #1004	Miner.PenalizeFundsInPriorityOrder: account for fromVesting greater than target	
#1007	Closed in #1004	Miner: Treat initial pledge like precommit deposits	
#1040	Closed in #1042	Miner: explicitly specify whether or not faults were added in RecordSkippedFaults	
#1064	Closed in #1140	Miner: Check ExpirationSet invariants	
#1068	Closed in #1159	Miner: Check Partition invariants	
#1070	Closed in #1158	Miner: Check Deadline invariants	

3.3 Priority 3

specs-actors Issue	Status	Title	Comment
#474	Open	ConfirmSectorProofsValid: batch VerifyDealsOnSectorProveCommit	Originally opened as #904
#667	Open	StorageMarket: Refund clients remaining balance since sector became faulty on termination	Originally opened as #694
#696	Open	Reward: Initialize `penalty` directly with the min value	
#721	Open	Recommendation: Standardize power/market cron method names	
#732	Open	PaymentChannel.Collect: Switch to “pull” payment pattern rather than “push” payments	
#799	Open	Power.OnEpochTickEnd: preempt Miner queries to Power and Reward	
#807	Open	Miner.Constructor: Additional input validation	
#905	Open	Miner: Simplify sector number allocation during precommit	
#913	Open	Miner: Drop precommits from prove commit set if power/pledge/reward values aren’t sane	
#980	Open	Power.CreateMiner does not allow caller to initialize Miner’s ControlAddresses	
#1002	Open	Miner: Unlock vested funds in SubmitWindowedPoSt	

specs-actors Issue	Status	Title	Comment
#607	Closed in #639	Market: ComputeDataCommitment should use ReadOnly, rather than Transaction	
#653	Closed in #830	StoragePower: Incorrect assignment to Cid fields during construction	

3.4 Other

These issues were not assigned a priority level for various reasons. For further details, see the provided issue link.

specs-actors Issue	Status	Title	Comment
#1090	Open	Reward: Check miner code CID	
#1144	Open	Market.PublishStorageDeals griefing vector	Known issue
#727	Closed	Verifreg.UseBytes: `newVcCap` is permanently lost if it's smaller than `MinVerifiedDealSize`	
#764	Closed	Power.OnConsensusFault: Assert `pledgeAmount` is non-negative	Method deprecated
#801	Closed	Miner.ReportConsensusFault: possible incorrect constraints on faultAge	Non-issue
#803	Closed	Miner control functions should abort on no-op	
#804	Closed	Miner control functions should abort for empty params	

specs-actors Issue	Status	Title	Comment
#805	Closed	Miner: NewDeadlineInfo may calculate negative Challenge and FaultCutoff epochs during first ~70 epochs	
#903	Closed	Miner: Remove references to variable seal proof types	
#918	Closed	Miner: bubble up error in processEarlyTerminations	
#955	Closed	Miner: Remove state.Transaction where not needed	Non-issue
#978	Closed	Miner: Correct for potential overflow when iterating over multiaddresses	Non-issue

3.5 Followup Work

specs-actors Issue	Status	Title
#1233	Open	Miner.Partition: Check sector existence on expiry and termination
#1234	Open	Miner.handleProvingDeadline: Check that method is being run at the correct time
#1235	Open	Miner.ExpirationQueue: Additional invariants
#1236	Open	Miner.ExpirationQueue: Enforce methods are passed unique sets of sectors
#1246	Open	Multisig: Allow proposals of batches of calls to enable complex actions
#1250	Open	Miner Policy: Correct truncating division
#1253	Open	Miner: Add balance invariant checks to cron methods

Appendix 1 - Related Links

The Master Tracking Doc was used to coordinate efforts between Consensys Diligence and Actors devs. It was shared with Actors devs in the first few weeks of the engagement, and was maintained by Diligence throughout. This document contains:

- Open Items: Bugs, recommendations, discussion items, and other relevant outputs were filed and tracked here.
- Schedule: Lists the primary and secondary objectives for each week of the engagement.
- Documentation: References the documentation we produced for each Actor in order to aid our review.
- Addressed: After addressing Open Items, issues were moved here.

Appendix 2 - Disclosure

ConsenSys Diligence (“CD”) typically receives compensation from one or more clients (the “Clients”) for performing the analysis contained in these reports (the “Reports”). The Reports may be distributed through other means, including via ConsenSys publications and other distributions.

The Reports are not an endorsement or indictment of any particular project or team, and the Reports do not guarantee the security of any particular project. This Report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. No Report provides any warranty or representation to any Third-Party in any respect, including regarding the bugfree nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on the Reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this Report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. CD owes no duty to any Third-Party by virtue of publishing these Reports.

PURPOSE OF REPORTS The Reports and the analysis described therein are created solely for Clients and published with their consent. The scope of our review is limited to a review of Solidity code and only the Solidity code we note as being within the scope of our review within this report. The Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty.

CD makes the Reports available to parties other than the Clients (i.e., “third parties”) – on its website. CD hopes that by making these analyses publicly available, it can help the blockchain ecosystem develop technical best practices in this rapidly evolving area of innovation.

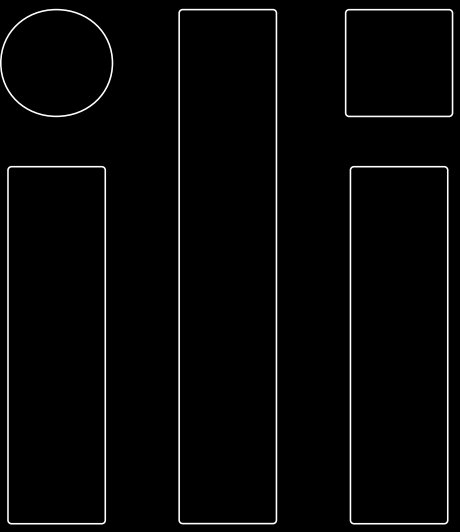
LINKS TO OTHER WEB SITES FROM THIS WEB SITE You may, through hypertext or other computer links, gain access to web sites operated by persons other than ConsenSys and CD. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites’ owners. You agree that ConsenSys and CD are not responsible for the content or operation of such Web sites, and that ConsenSys and CD shall have no liability to you or any other person or entity for the use of third party Web sites. Except as described below, a hyperlink from this web Site to another web site does not imply or mean that ConsenSys and CD endorses the content on that Web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the Reports. ConsenSys and CD assumes no responsibility for the use of third party software on the Web Site and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

TIMELINESS OF CONTENT The content contained in the Reports is current as of the date appearing on the Report and is subject to change without notice. Unless indicated otherwise, by ConsenSys and CD.

Request a Security Review Today

Get in touch with our team to request a quote for a smart contract audit or a 1-day security review.

CONTACT US



AUDITS

BLOG

TOOLS

RESEARCH

ABOUT

CONTACT

CAREERS

Subscribe to Our Newsletter

Stay up-to-date on our latest offerings, tools, and the world of blockchain security.

e-mail address