

Flare Network

Smart Contract Audit



Flare

Smart Contract Audit

V230328

Prepared for Flare • March 2023

1. Executive Summary
2. Assessment and Scope
3. Fix Review
4. Summary of Finding
5. Detailed Findings

FLR-6 Reward burner can be reconfigured to send funds to arbitrary account

FLR-7 priceDeviationThresholdBIPS can't adapt to market conditions

FLR-8 totalSelfDestructWithdrawnWei variable is never updated

FLR-9 Inflation annum tests not passing

5. Disclaimer
6. About Coinspect

1. Executive Summary

In July 2021, [Flare Network](#) engaged [Coinspect](#) to perform a source code review of the Flare Network smart contracts.

The objective of the project was to evaluate the security of the smart contracts that implement critical functionality in the network, including but not limited to:

- Flare Daemon
- Inflation and Supply
- Flare Time Series Oracles and Price Providers whitelisting
- Vote Power Token and delegation mechanism

In February 2022, [Flare](#) engaged [Coinspect](#) to perform a second source code review of the **Flare Network platform**. This latest audit focused on the incremental changes performed to the platform since Coinspect's previous review.

Overall, Coinspect observed Flare smart contract's code, documentation, and test suite quality to be above average. The code follows smart contract security best practices. All the reviewed components had clear specifications and the code was extensively documented with commentaries explaining relevant decisions and assumptions. Several unit and integration tests are included in the repository and result in excellent code coverage.

Coinspect found no issues that could result in user funds being at risk. Some informational level findings are reported as suggestions to further improve the platform.

In March 2022, [Coinspect](#) performed a Fix Review of the **Flare Network platform**, to reevaluate the security of the smart contracts in scope.

No issues were identified during the assessment:

High Risk	Medium Risk	Low Risk
Open 0	Open 0	Open 0
Fixed 0	Fixed 0	Fixed 0
Reported 0	Reported 0	Reported 0

The engagement focused on the platform’s smart contracts and their correctness with respect to the specifications. The design of the platform’s crypto economic incentives and related assumptions were not reviewed as they depend on the interaction with other components which are not yet finished and were not part of this engagement. The Flare Network’s consensus layer and Governance model were not evaluated either.

2. Assessment and Scope

The latest audit started on **January 24, 2022** and was conducted on two branches of the GitLab repository located at gitlab.com/flarenetwork/flare-smart-contracts/.

The first branch reviewed was [coinspect_audit2_branch1](#), which included the changes for the Songbird candidate release. The last commit reviewed during the engagement was `e3fa5c15421707b3bc85327b3442e856d2ba5e60` of **January 24, 2022**:

```
commit e3fa5c15421707b3bc85327b3442e856d2ba5e60
```

```
Merge: 9d8c9c1 9e0a422
```

```
Author: Ilan Doron <ilan@flare.network>
```

```
Date: Mon Jan 24 09:20:23 2022 +0000
```

```
Merge branch 'songbird-candidate' into 'coinspect_audit2_branch1'
```

```
remove some outdated docs
```

```
See merge request flarenetwork/flare-smart-contracts!423
```

The second branch reviewed was [coinspect_audit2_branch2](#), which included the changes introduced for the Flare candidate release. The last commit reviewed during the engagement was `f774e1d2ac90634feb6f12c303793db41bb4a419` of **February 8, 2022**:

```
commit f774e1d2ac90634feb6f12c303793db41bb4a419
```

```
Merge: 5792771 43f57b8
```

```
Author: Ilan Doron <ilan@flare.network>
```

```
Date: Tue Feb 8 17:00:29 2022 +0000
```

```
Merge branch 'master' into 'coinspect_audit2_branch2'
```

```
update from master to coinspect audit 2 branch.
```

```
See merge request flarenetwork/flare-smart-contracts!445
```

The scope of the audit was limited to the following Solidity source files, shown here with their sha256sum hash corresponding to the second branch reviewed:

```

bf4e539e6ec09b9d9472efa79923b4a7d7232cc5f119206c74d52d6ae841dccc ./tokenPools/implementation/FtsoRewardManager.sol
68e536215e580bc0073a401087a09f0921f2a75c9c62e7295ca1803cf31506f1 ./tokenPools/implementation/UnearnedRewardBurner.sol
a0e52c0e5cc716c4aa158b2d0b6d9d48726858139f847228faa38f6d9c3366f3 ./tokenPools/implementation/Distribution.sol
8cea4c20f387577b2cea18c134cf8520b3e022184f33c84eb7c7f54084919e937 ./tokenPools/interface/IITokenPool.sol
239b1a4cea7f3097af27803b9abdd97d26bd5bf126abbcee2f2b9e6e2610888c ./tokenPools/interface/IIFTsoRewardManager.sol
ea8a2ae0d3a5355929103d1006bd9e33bc361b5c6193e3b5749448383c4ad53 ./mockXAsset/mock/AssetToken.sol
a74a32cc14ad2134c626485068e5c76b3071bc5057ced810a966757745bfe329 ./mockXAsset/mock/DummyAssetMinter.sol
e048e4b7718b3070b7bce54987b84330318703286ddd0c79d7ab738515eac52f ./mockXAsset/interface/ICollateralizable.sol
aea5d38cb92435e56b8601b5f3fd7b5e5062da993feebd249c71b9bb6481f1a8 ./genesis/mock/InfiniteLoopMock.sol
6b40d53e3dfbc29c43aa039797d59f61773b4ace0053c3f800157bab8ce808a0 ./genesis/mock/FlareDaemonMock.sol
26b79fc0fce5056ba4008f0536b01525aacd414c6dbcabac429163b7b3de375 ./genesis/mock/TestableFlareDaemon.sol
debdeba7884368523c7022e3e73731cfc5aa5e53f24c76163b692a3eb5a5ae99 ./genesis/mock/FiniteLoopMock.sol
711f9844fc81676a95498cd1d6497cd125fa6627ba98dc52ce987add9577900 ./genesis/mock/FlareDaemonMock1.sol
6a0183517977afc120b627432f8dfc6d8b5eb2be17d312ef791c2d3b803908e9 ./genesis/mock/ReadGasLeft.sol
2060606587d0518f3397e3bc0d7aa582b1c0eb4bbdf7b64cc2e339bab61d2ccb ./genesis/mock/EndlessLoopMock.sol
3f0cc4a73ff5cba07c18186a9cb44558e348f5a43d243facd8a96b706e1b34d0 ./genesis/mock/FlareDaemonMock2.sol
3f6eff7ec1ab34756f26e822a7c17eb24578a35ac2c2bafee013b9e438e20db0 ./genesis/mock/InfiniteLoopMock1.sol
3ad83e36cf28d87b6e5f606f520665ff228d1c9132c7ee20727b22de443c1278 ./genesis/implementation/StateConnector.sol
296270f8983befd1d8234e8675ce33f7eb9489b578c5c5d1fbdaba541fd20045 ./genesis/implementation/PriceSubmitter.sol
953732bfa7eeaed316ed77eaf1fcc167eae65e10c75248ee52eb37350b22cea ./genesis/implementation/FlareDaemon.sol
435152b7625dbc1564bb275e819e11d65f540d183361a04061406eaeacf19f00 ./genesis/interface/IFTsoRegistryGenesis.sol
90cac97dd9882b6f77e4a1a90dd9a812c10d8266b41c6287a8d4285d6cad0314 ./genesis/interface/IInflationGenesis.sol
4352cd4e2d3644b5ba19b261a6b0c65d804dc0f4fe2c64910c5aa88ffc358430 ./genesis/interface/IIPriceSubmitter.sol
cca145c8c67706995747db27662f1d76eb33d6b54a0315fdac396fae4747c76 ./genesis/interface/IFlareDaemonize.sol
06a7b4f3c2b67a0a92094f6e53ced08da3f8abadd3859f1e1bfcf8e70fe8c05 ./genesis/interface/IFTsoGenesis.sol
907c5c8b95ff8dcccdd8572c63654716fe9f1d626df43520f1a5995bbd388019f ./genesis/interface/IFTsoManagerGenesis.sol
888dee1a4d362fcb2c0f154f56116dec520f0b23e126942f52813a32aa790811 ./addressUpdater/mock/AddressUpdatableMock.sol
fb74132581784ebcd120221c5478b26ddb915bb0a5707856cc52bf50412d1770 ./addressUpdater/implementation/AddressUpdater.sol
6cf01dd4544e028cd562bb0cc9d9c1f1c350b9ae2fc6dc3d96b3af28f20fd095 ./addressUpdater/implementation/AddressUpdatable.sol
59824abf5524c24e54c161f1faeed77763eab397b4eb7d02304291fbc63d4abc ./addressUpdater/interface/IIAddressUpdatable.sol
f1b03df0e5b8863aed1cb97c5b0ec7bef51837713c14f18b4a64e3974f01cbe ./governance/mock/HistoryCleanerMock.sol
fb582eaf980b1adb4d78ebf6a83e5a522e68fa990f961feda28291dfca6fea5a ./governance/implementation/GovernedBase.sol
93cf605f3233de4520b4e27474677166106690aa8e2563e135d5d98720356700 ./governance/implementation/Governed.sol
e0d147fb7158a7a866ccae28bca80296f032cd8db1e9fc40e8262a725a8bc7eb ./governance/implementation/GovernedAtGenesis.sol
4d30ff003e16b3a625fbd20ecc39965771ff9efdf3a3a0edb87f14be80b4344 ./userInterfaces/IFTso.sol
22b2f694b7bf7eb2fd447785b7434346a77003edb9db6fdaa2ea79932debff4 ./userInterfaces/IVPContractEvents.sol
fd3111ed7cbaa101d4bb85eb79e05469c6ad20355f66149b15a2935002b44cb7 ./userInterfaces/IPriceSubmitter.sol
bddea7532299880f0a76e9067266d4fa18477826096326c7c1bf448e84c6a0b4 ./userInterfaces/IWNat.sol
b850523a4bf8173ca747030b4db8a9118ef2992a7264f6afbbf509623c0aa13c ./userInterfaces/IVToken.sol
39e3e7de96bc794f830dad2c2c2c086ad9aeac1d1e886934679d7cc3d315f95c ./userInterfaces/IDistribution.sol
482aa07addf41346e3996b31ed592c7cba28b95e0f0ebd0ebeeafae79b8f8ed ./userInterfaces/IFTsoRegistry.sol
6cb0ad319bf05b26a4cf846273c3db2394c83241612bfad57fb42cab19cba1578 ./userInterfaces/IGovernanceVotePower.sol
9f80875e92cb3067aab251ad746136b3e3f55c5fa9f129643132fb487196b35 ./userInterfaces/IFTsoRewardManager.sol
43d8624273b06f2ef77a277b5ad98b1c3fe7ae89edf6c627711a26e4b2b94074 ./userInterfaces/IFTsoManager.sol
eaa082812a3486b88d44bc25c1ead80fc0d3993dfa7c4c2669c0842c49245d9 ./userInterfaces/IVoterWhitelister.sol
27235ab0ac08281c05f39130bf12ad8237cd5850fee2fd64bfba200b673e876 ./inflation/mock/InflationReceiverMock.sol
c251acd5490b7e08e5fdeedd0ae2ab4fab0905645a0455f2407af85e55c00896d ./inflation/mock/InflationMock.sol
5e17fe844f3db062e3bfa397254ba17b6cc6e56f5d6b3462073941b671a9ae4a ./inflation/mock/PercentageProviderMock.sol
483d4d4e050ebcf4dfbc04a9e20a0a7fdaf287d920c98ea128713d4727af9af ./inflation/mock/SuicidalMock.sol
04b20a9bf8e179038d98d15711f1cb37c9292e5aff538f8d2022c40af8511a52 ./inflation/mock/InflationMock1.sol
77e4fd4021fae07fa17b34b026293f2ce76d6e076c665bbf25d7dc22b3c646f0 ./inflation/mock/FlareDaemonWithInflationToken.sol
234730acd6c7a35fb1d69672c164c6d6360e738544dffd8f21e7b254266a91c3 ./inflation/mock/InflationReceiverAndTokenPoolMock.sol
f63c5c572d823d6c978b875f99cb2156bbc0080f113ff94aaebaf97048c7c95a ./inflation/implementation/Inflation.sol
d9f68dbb03cbad352829cd72519076618a56c6d777c1f7b2949420bbc36edcbcb ./inflation/implementation/Supply.sol
40dda7efa90cd1db1d8f813205b718b9dbd8d0e9057375dff37a1ef7b5d80f4 ./inflation/implementation/InflationAllocation.sol
3b37472bebf05518c1cde004da4133afeb4272c07e31d8db317f13416c78dbca ./inflation/lib/InflationAnnum.sol
d0df108753a5d4aa4351266a547f2db3ea27019ac9a18034d9b58f443c167cf9 ./inflation/lib/InflationAnnms.sol
942dd3f1315811deb28deb14dfdaf47225e396733bc99b3391f4e240f5d7e3d2 ./inflation/lib/RewardServices.sol
b87a9d8f63b2ef7cd0f507339d6f2a6497378a84940269f8156d24597adec71 ./inflation/lib/RewardService.sol
a29709dcd968b1d91cfba5e354eb8f54314668e834de300dc69a5187bf9b15 ./inflation/interface/IIInflationAllocation.sol
92f6f58dd0c54c6b1ac545a0e24479acef5673b21f3ebc142f68ff3a3ebe9ad ./inflation/interface/IISupply.sol

```

```

093ce091266460e8f4d4682c3d235ae63ae20590dd3f1fd052ee03c752586ac ./inflation/interface/IIInflationReceiver.sol
1f19fc317f93cadaaf8937db4fcb61aa17f4949694e7cec9a7ab958278c31733 ./inflation/interface/IIInflationPercentageProvider.sol
32a518978197b162f36bbcee446200d6424e126c8077b93ee9c7ace449a7ad0d ./inflation/interface/IIInflationSharingPercentageProvider.sol
b13a5fe247a4b80022cbeb5a63f371de391f3a6da8947476075d7a7c95abf8d9 ./utils/mock/PriceSubmitterUnregisterHack.sol
8987b038958572bbd87aee6862ac2ec3770aa158037a1ff4523767b3bf1691af ./utils/mock/SafePctMock.sol
ef603d230385fbefec6d97312ac1f2e9d73beed653654e04cbb942b1f3b6fc1c ./utils/mock/VoterWhitelisterMock.sol
6945f0b2926288db2a947c91fd30fec36f57373423853b47c5f825099cda3dc ./utils/mock/GasConsumer3.sol
6da309494640bd043d894407fce4c7dae1948f8c289a8b66afac5e2587d027cc ./utils/mock/GasConsumer.sol
e2e45ea0659ab4b848ba5fb9f79c99d519db95da524901232cdd96670ea3c616 ./utils/mock/GasConsumer5.sol
ec1a3852a028d30cc2e8ff18206110b46fcc1ee1f345a7e512e28f0696c94baa ./utils/mock/GasConsumer7.sol
83da6848fc51feec6b3165f7f2917d74eb3dc19b7ea32262b13d7c2322edc5ef ./utils/mock/GasConsumer2.sol
f90c7c91a6c17c2fa1a5821671fd21e5fb303a79721709cfcfd70ff88591d849b ./utils/mock/GasConsumer6.sol
0ea9b26b1502d7507ed8a9ddeae625718d24b3830749d80c71a077393541e01fc ./utils/mock/DateTimeLibraryContractMock.sol
7456379c2cc2e3ba2856c7798ed6ab6e4f802030cda14e4dbf4ac55b04d4b30d ./utils/mock/GasConsumer4.sol
0aa255d92f4f87c1d4ad4066574e08bfd9350dd0352f6188d9680ee0c072458b ./utils/implementation/PriceReader.sol
c3fd3d454ce3d09c90c22c58631febdb2e5b17499e2c269775f393d5675a59e7 ./utils/implementation/VoterWhitelister.sol
2c134d5f194559ad9e087297bbd669a22c20c7f76aaf93bde833e262682795c ./utils/implementation/RevertErrorTracking.sol
da7d13381925ca65b5b184d651fb0a5af86a7146b4bf5b8735c64a51bc51bb68 ./utils/implementation/DateTimeLibrary.sol
0dacb948c6a6011d0493f9e8992cadf09080bb6d5773636dfdad85562c5f42bf ./utils/implementation/SafePct.sol
bce891791f2cf868f98158ee333e41c53df1ebb138c661dac836472386c56512 ./utils/implementation/FtsoRegistry.sol
a81c7e3cbda4eb930d004f48745a7030c35a694fd78579b416c31d8f6f7d43ef ./utils/implementation/GovernedAndFlareDaemonized.sol
3617f640c89459bb1f020fbc0f6f4adc0e2c574735970af0586b651618661ac4 ./utils/Imports.sol
37c6eafd11bbcb296df58c3234410ca756617aa61506b188d43c73db8230d4b0 ./utils/interface/IIVoterWhitelister.sol
d7971865bc3504135040b25015c8204e41ca8d3ed4209e096dd21e9f7458b04f ./utils/interface/IIFtsoRegistry.sol
61ef59581eaf0270928c91317ebaf15f911e7c07ce34c8ec72594692c9a0ee3 ./ftso/mock/FtsoMedianMock.sol
933d960e83ec96ffa01ca049ce67850230eebdad3db880f8016eb25020f223f3 ./ftso/mock/MockFtso.sol
1245adac9a228015d4a6add6c999c2d6e978874593eb08b85163daabba88ea ./ftso/mock/FtsoManagerMock.sol
5f813d421bc347975d69feb95a924a49d828e95839b8d0110061e7a96a42834e ./ftso/mock/MockVPToken.sol
9b17cd3231747b5994aeb0c4df08f2768bc1f201d62d62398525d8c05b0514b8 ./ftso/mock/FtsoEpochMock.sol
85cddafd066db0ba9aa6bc79672c918d0e64907c5573f6b3b5fc0437aec81264 ./ftso/mock/FtsoManagerV1Mock.sol
e57aff43aeb20daf18ccae832c0dbd024d0ec838d46d7e7943a0e4f1d2fbba47 ./ftso/mock/SimpleMockFtso.sol
e6c05d055ccbebcba1038124f1d1e29b19fe737d1f5842e5a52f8a36a433be7 ./ftso/mock/FtsoVoteMock.sol
5067e9001be2fb8e3c18aa7c5ff0b0efea84c74bdfbc887926d4606b723eebd ./ftso/implementation/Ftso.sol
6569e950d567c5bbf820776331ad87b0d7720bda42bacd40c899eaf474c994 ./ftso/implementation/FtsoManager.sol
5518904b8925acfb093b46b5fda5b29578fd2987d27f3de9d6b21813ba068d2d ./ftso/priceProviderMockContracts/PriceProviderMockContracts.sol
27aa11517762ff22077443b1b8d76eb617ee123f720b409d192f75e654094afe ./ftso/priceProviderMockContracts/priceProviderMockFtso.sol
6147dd29e7b0ad02bb1351020d9fbc0079b6091f948b43f8a7c4c132a03ba5f1 ./ftso/lib/FtsoVote.sol
5fe21742a36e432bcdce329240ef8621439c7cc2360f80722d3495d967539cd ./ftso/lib/FtsoMedian.sol
9fc897b5beaa451a98ce4d1e5680763926f544b92285835744749ccdc6c284abb ./ftso/lib/FtsoEpoch.sol
610ffe35ef8d0f7635546da682e0b85066f6d7da531e42b521550a8823143feb ./ftso/lib/FtsoManagerSettings.sol
bfd77f3cd9a85464aef72a352738d125aa8ccc44b01151177f500f45824b24e1 ./ftso/interface/IIFtsoManager.sol
7453e1b5f1ff09150633a05d0d3f3933b0305e36726dce6dff3fad90cf34e05c ./ftso/interface/IIFtso.sol
c5cdbdf3da71c72202233f06cc0cfd484adb064add6b467a8e2818212ba13ad9a ./ftso/interface/IIFtsoManagerV1.sol
795ed7e1637f23eacf1a80da16aa2eb4785e65d2f9cac0c4d89a6b3f651fa9bf ./token/mock/VPTokenMock.sol
f9a26c45723f722eb7fb92c03559b23f20393559b0a3a4be2bc552f189757a ./token/mock/DummyVPToken.sol
dce2c78c18163d8540271e4a4b74cafe680e80f8ac853442a81d847b7b5099c5 ./token/mock/TransferToWnatMock.sol
4c94b4f56769094663535f0d42148691e5736ceaea8e87fcd59bd2f1cb5b919c ./token/mock/FlashLoanMock.sol
841498726cdf0115b452cf8f35a657a6ee2d21a1e775a88ee7ea7f733b9b33fff ./token/mock/CheckPointsByAddressMock.sol
cea31e015368633710f9857b298c7f1a6fecfee141c2a30c4cd5cdf2d8e98e532 ./token/mock/PercentageDelegationMock.sol
666fe341bf0b0a5f0e28d7d339581eb89adb117fc095f99bd66d7dc37b3a5940 ./token/mock/CheckpointHistoryMock.sol
9f9022ac627b70ba075e57bb1aa978bbcb3b4d07ceb9be7e2acf075294a4e41 ./token/mock/VotePowerMock.sol
8a06b88a399e75a55a404fb8fbc970db8643f6b47a5ce4ab24a306dc1eb4d3c ./token/mock/ExplicitDelegationMock.sol
4246d9082408e18033c1d2d5d6776242878a4cdeca24c696dd7e787a8f7fa8ec ./token/mock/VPCContractMock.sol
85476a8b8d03782f81331edbd527513f70b2d887222f68339d66fd7729d9057 ./token/mock/DelegatableMock.sol
91bdd6388216144771835c99cfff1c6974d35d36ee5d1fb5a017066ee835b6c86 ./token/mock/CheckpointableMock.sol
cdf463a2c26f7245f3678565cb1275a5ffef6d633fa329e19c1e84e6af333d853 ./token/implementation/Checkpointable.sol
04a3bc74d57bbf9a84cabfa308b8381bd225db7a210a579f01319e6a3f89ffff ./token/implementation/VPToken.sol
5003b4781ad9ad55718dbb6607ec6691078a8994779c15cf507fec52565e4a7e ./token/implementation/Delegatable.sol
aff920a71e29771dab9fedbd55614b303a21aa18b9bd2fe9a2fd0c3e3ca3426b ./token/implementation/WNat.sol
6b18c3c39097c659979825df30f912e80f4e9bd518e0b8b149d0ed8b4836c1dc ./token/implementation/CleanupBlockNumberManager.sol
23f612bdabff273dbff98deadca84fe9fba7e3f2ab31c6804484f2a6904200c ./token/implementation/VPCContract.sol
53aa2327f50bb17d650419207735c18c7e90bddddef074bc80e240c1442d01 ./token/lib/DelegationHistory.sol
796d9b62af94c15d89b94676d4514396e8a8c165c9e870e099996b3a0488cd63 ./token/lib/CheckpointHistoryCache.sol
7c7858e64134e57a649b54e5e04153f448d346e246b84923d4339c4fb576751f ./token/lib/ExplicitDelegation.sol
82da0011307842f905d1b3809457ad58a0e869b7c0590a2edab9f6011e1f3c35 ./token/lib/PercentageDelegation.sol

```

```

ed07ecbf063aafb852b4e67baf68d7a02485144db31ed37765cdab9b9e2396bf ./token/lib/VotePowerCache.sol
8ec06551b24efb1a3aef75ec0af76eca1f2845bad71eeafadd71ed514c0ab116 ./token/lib/VotePower.sol
8b964e7efa5b4ddb469251fb8b455f65964ed187f82707ff7eadc089ed41f8e6 ./token/lib/CheckPointsByAddress.sol
10f17a1c451bc43f55ba0a8a81a9a213dc1df6ec2bb7200cc21d13f422ed34dd ./token/lib/CheckPointHistory.sol
63856382a0837b1e219b5f5e49a402a90877d1e4b9d3469ade5add2613fd4b8a ./token/interface/IICleanable.sol
ad7bd71fb0f01dea138ff1fb0213100248873450301a2ad674285ba8cf56b05d ./token/interface/IIGovernanceVotePower.sol
c3f1fdbd8e3b1c298552586f4c07fc1bdd28c45d1d30dd765c3c0443afab3ca6 ./token/interface/IIVPContract.sol
37d9c059841c5c00853b6407cbb82cd15a8d5e144e7a0ecc039c7f818c0ccd94 ./token/interface/IIVPToken.sol

```

The following documentation was consulted during this engagement:

1. Flare design review presentation prepared by Flare’s team
2. The Flare Network and Spark Token whitepaper [flare_v1.1.pdf](#)
3. System specification documentation
https://gitlab.com/flarenetwork/flare-smart-contracts/-/tree/coinspect_audit2_branch2/docs/specs
4. Flare Consensus Protocol [FCP.pdf](#)

The consensus protocol used in Flare network, the attacks that could arise because of its particularities and the design of the economic incentives or lack thereof were not in scope for this audit.

This audit focused on several smart contracts that implement system critical functionality for the Flare Network such as:

- Token contracts
- Flare Daemon, a special system trigger contract.
- Flare Inflation tracking and allocation.
- Reward manager and distribution to rewarded services
- Supply accounting system of FLR tokens
- Vote Power (delegation mechanisms and revocation)
- FTSO (Flare Time Series Oracles)
- FTSO Manager
- Price providers and the whitelisting process

A detailed explanation of how these components work and interact can be found in Flare Network’s [website](#) and project repository and the material listed above in this report.

The system's specifications include attack scenarios and how they were contemplated and mitigated and the rationale behind several choices made for the implementation.

The smart contracts are specified to be compiled with Solidity compiler version 0.7.6. The repository includes a comprehensive set of unit and integration tests. The code coverage obtained is excellent, except from one contract that is part of a new feature that is being developed (`StateConnector.sol`). Also, the code repository includes a set of fuzzers that test tokens behavior and price providers and their voting power as a random sequence of protocol actions occur.

The contracts are designed to be upgradable, and most of their parameters can be reconfigured after deployment by a special Governance address controlled by the Flare Foundation. The Governance mechanism responsible for setting these parameters, and as a result controlling the platform and its funds, was not evaluated by Coinspect.

During this engagement, several potential attack scenarios were considered and evaluated by reading the code and writing tests, including:

- Pseudo random number generation and utilization
- Potential issues with vote power delegation
- Revocation of vote power delegated in the past and its related cached view
- Price providers collusion to manipulate prices
- Price providers collusion to unfairly accumulate rewards
- Transactions replay and the commit/reveal scheme used to submit prices to oracles
- Accounting of balances maintained by the system contracts
- Bypassing inflation and supply token minting limits
- Data structures and gas utilization abuse

For this audit, Coinspect auditors assumed the security properties granted by the Flare Network consensus layer behaved as described in the whitepaper. More specifically reorganizations and the manipulation of transaction ordering were considered impossible. It is advised to consider the possibility of attacks relaying on protocol related time windows and how they could be manipulated. For example: would it be possible to spam the network with enough transactions right before the

reveal period starts in order to force the commit transactions out of the time window, censoring late submissions in order to manipulate the prices? (this could be facilitated by the price providers' late submission pattern as observed in Flare monitoring tools). The profitability of such an attack would depend on the value of the costs being protected by the FTSO oracles and this was outside the scope for this engagement.

Even though the cryptoeconomics and incentives mechanisms were not evaluated during this engagement, which focused on the implementation correctness, some high-level concerns were identified during the code review in relation to oracle manipulation, price provider collusion scenarios, and incentives alignment and were discussed with Flare's team. These potential issues depend on other components of the platform which are currently being developed and will be considered by the Flare team and it is recommended they are reviewed once the implementation is complete.

It is worth noting that several protection mechanisms are implemented in the platform, including but not limited to: configurable parameters (e.g., maximum allowed price deviation), monitoring infrastructure and fallback to trusted addresses. Most of these mechanisms depend on the ability of external observers and off-chain components to identify behaviors and react in a prompt manner.

Regarding the oracle consumers, Coinspect recommends adding the ability to obtain the epoch finalization type besides the current price and timestamp from a FTSO oracle `getCurrentPrice` function in order to allow the clients to make decisions based on how the price was calculated (e.g., fallback mode, price copied from previous epoch) without requiring them to listen to the blockchain events.

3. Fix Review

The Fix Review started on **March 24, 2022** and was conducted over the files located at the [coinspect_audit2_branch2](#) GitLab repository, which was [merged](#) on **March 15, 2022**, as of commit `42ff03c74f10f816e606d02f2fb6b741a3400195`. The files have the following sha256sum hash:

```
bf4e539e6ec09b9d9472efa79923b4a7d7232cc5f119206c74d52d6ae841dccb ./tokenPools/implementation/FtsoRewardManager.sol
68e536215e580bc0073a401087a09f0921f2a75c9c62e7295ca1803cf31506f1 ./tokenPools/implementation/UnearnedRewardBurner.sol
a0e52c0e5cc716c4aa158b2d0b6d9d48726858139f847228faa38f6d9c3366f3 ./tokenPools/implementation/Distribution.sol
8cea4c20f387577b2cea18c134cf8520b3e022184f33c84eb7cf540884919e937 ./tokenPools/interface/IITokenPool.sol
```

```

239b14cea730954f27803b9abdd9726bd5bf126abbcee2f2b9e6e2610888c
ea8a2aed03a5355929103d1006bd9e33bc361b5c6193e3b5749448383c4ad53
a74a32c14ad2134c626485068e5c76b3071bc5057ced810a966757745bf5e329
e048e4b7718b3070b7bce54987b84330318703286ddd0c79d7ab738515eac52f
aa5383cb92435e56b8601b5f3f7d7be5062da993feebd249c71b9bb6481f18a
6b40d53e3dfbc29c43aa039797d59f61773b4ace0053c3f800157b88ce080a0
26b79f0cfe5056ba4008f0536b01525aacd414c60bcababc429163b7b3de375
debdeba7884368523c7022e3e73731cfc5aa5e53f24c76163b692a3eb55ae99
711f9844fc81676a95d498c1d6349c1215fa6627ba98dc52ce987add9577900
6a0183517977afc120b627432f8dfc6d8b5eb2be17d312ef791c2d3b803908e9
2060606587d0518f3397e3bc0d7aa582b1c0eb4bbdf7b64cc2e339bab61d2ccb
3f0cc4a73ff5c0a07c18186a9cb44558e348f5a43d243facd8a96b706e1b34d0
3f6eff7c1ab34756f26e822a7c17eb24578a35ac2c2bafee013b9e438e20db0
6e8bb7838a86dd570b75344de42649046fc160b985944955f18b2b773a23e67f
f108624d70a9569b936547e4dcf0a7bb5318c68bafced29d8a0a631b34e5f3
953732bfa7eead316ed77eaf1fca167ee65e10c75248ee52eb37350b22cea
43512b27625dbc1564ab2758e19e11d65f540d1183361a04061406eaeac19f00
90cac97dd9882b6f77e4a1a90dd9a812c10d8266b41c6287a8d4285d6cad0314
4352c4de2d3644b5ba19b261a6b0c65d804d0c4fafe2c64910c5aa88ffc358430
cca145c8c67706995747db27662f1d76eb33db54a0315fdac396fae4747c76
06a7b4f3c2b67af0a92094f6e53ced08da3f8abadd3859f1e1bfc8e70fe8c05
907c5c8b95ff8dcd8572c63654716fe9f1d626df43520f1a5995bbd388019f
888dee1a4d362fcb2c0f154f56116dec520f0b23e1269425f2813a32aa790811
fb74132581784ebcd120221c5478b26ddbb1915bb0a5707856cc52bf504121770
6cf01dd4544e028c5d562b0cc9d9c1f1c350b9ae2fc6d3d96b3afd28f20fd95
59824abf524c24e54d161f1faeead7783eab397b4eb7d0230429191fbc63d4abcc
f1b03df0e58863aed1cb97c5b0ec7be5f1837713c14f18b4a64e3974f01cbe
fb582ea6980b1adb4d78ebf6a83e5a522e68fa990f961feda28291dfca6fea5a
93f605f3233de4520b4e2744677166106690aa8e2563e135d5d98370356700
e0d147fb7158a7a866ccae28bcac0296f032cd8b1e9fc40e8262a725a8bc7eb
8fcef6d8cb170c9c9d9d2c2c2c086ad9aeac1d1e886934679d7cc3d315f95c
482aaa07addf41346ce3996b31edd592c7c7ba28b95e0f0ebd0ebefae79b8f8ed
6cb0ad319bf05b26ac4f846273c3db2394cc83241612b2fad5fb742cab19c1a578
9f80875e92cb3067aabb251ad746136b3e3f55c5fa9f129643132fb487196b35
43d8624273b06f2e77a277b5ad98b1c3f7ae89edf6c627711a26e4b2b94074
eaa082812a3486b88dd4cb2c2c2c086ad9aeac1d1e886934679d7cc3d315f95c
27235ab0ac08281c05f39130b12ad8237cd5850fee2fd64dbfba200b673e876
c251ac4590b7e085fded0ae2ab47fb0905645a0455f2407af85e55c008962
5e17f8e44f3db062e3bfa397254ba17b6cc6e56f5d6b3462073941b671a9ae4a
4834d44e050ebcf4dfbc04a9e20a0a7fda7287d920c98ae128713d4727af9af
04b20a9bf8e179038d98d15711f1c1b37c9292e5aff538f8d2022c40af8511a52
77e4f4021fae07fa17b34ab026293f2c76d6e076c665bbf25d7dc22b3c646f0
234730acd6c7a35f1d69672c164c6d630e6738544dfdd8f21e7b254266a91c3
f63c5c572823d6c973f8bce446200d6424e126c8077b93ee9c7ace449a7ad0d
baaf6e6f6ea5689ea0f469a3eb403839d3780b7e0d98dc7c124d04e65fb2f7
40dda7efa90cd1db1d8f13205b718b9dbd8d0e9057375dfdd37a1ef7db580f4
3b37472ebf05518c1cde004da4133afeb4272c07e18d8db317f13416c78dbca
d0df108753a54daa4351266a547f2db36e27019ac9a18034d9b58f443c167cf9
942dd3f135811deb28dbed14dfdaf47225e396733bc99b3391f4e240f5d7e3d2
b87a9d8f63b2ef7c3df05973396d72a6497378a84940269f8156d24597adec71
a29709dcdb968b1d91cfba5e354eb8f54314668e834de300cd69a5187bf9b15
9226f58dd0c54c6b1ac545a0e24479acfe5673b21f3ebc142f68ff3a3eb9e9ad
093ce09126640e8f4d4682c3d235a63e20590dd3f1fd0852ee03c752586ac
f1f9fc317f93cdaaaf8937db4fcb61aa17f4949694e7c9ca9a7958278c31733
32a518978197b162f36bbce446200d6424e126c8077b93ee9c7ace449a7ad0d
b13a5fe247a4b80022cbeb5a63f371de391f3a6da8947476075d7a7c95abf8d9
8987b038958572bbd87aae6862ac2ec3770aa158037a1fff4523767b3bf1691af
ef603d230385fbefec6d97312ac1fe29d73bed653654e04cbb942b1f3b6fc1c
6945f0b2926288db2a947c91fd30fec36f57337423853b47c5f825099cda3dc
6da309494640bd043d894407cdec47c1948f8c289a8b66afac5e2587d027c2
e2e45ea0659ab4b848a5f5b9f79c99d519db95da524901232cd96670ea3c616
ec1a3852a028d30cc2e8ff18206110b46fc1ee1f345a7e512e28f0696e943baa
83da6848fc51feec6b3165f7f2917d74eb3dc19b7ea32262b13d7c2322edc5ef
f90c7c91a6c17c2fa1a5821671fd21e5fb303a79721709cfcfd70ff88591d849b
0ea9b26b1502d7507ed8a9ddea625718d24b3830749d80c71a077393541e01fc
7456379c2cc2e3ba2856c7798ed6ab6e4f932030cda14e4dbf4ac55b04da4b30d
0aa255d92f4f87c1d4ad4066574e08bf49350dd0352f6188d9680e0c072458b
c3fd34d54ce3d09c90c22c58631febdb2e5b17499e2c269775f393d5675a59e7
2c134d5f194559ad9e4087297b6d69a22c20c7f76aaf93bde833e262682795c
da7d13381925ca5b5b184d651fb0a5af86a7146b4bf5b8735c64a51bc51bb68
0dacb948c6a601d0493f9e8992cadf09080bb6d5773636dfad85562cf542bf
bce891791f2cf868f98158ee33e41c53df1eb138c661dad836472386c56512
a81c7e3cbda4eb930d004f48745a7030c35a694fd78579b416c31d8f6f7d43ef
3617f640c89459bb1f020fbc0f6f4adc0e2c574735970af0586b651618661ac4
37c6eaf11bbcb296df58c3234410ca756617aa61506b188d43c73db8230d40b
d7971865bc3504135040b25015c8204e41ca8d3ed4209e096dd21e9f7458b04f
f759581eaf0270928c91317ebaf15f911e7c07ce34c8ec72594692c9a00e3
933d960e83ec96ffa01ca049c6e7850230eebdad3db880f8016eb250207223f3
1245adac9a228015d4ad6addcd6c999c2d6e978874593eb08b85163da4abba88ea
5f813d421bc347975d69eb95a924a49d828e95839b8d010061e7a96a42834e
9b17cd3231747b5994aeb0c4d08f2768bc1f201d62d62398525d8c05b0514d8
85cdda0f66db0ba9aa6bc79672c918d0e64907c5573f6b3b5fc0437ac8126a
e57aff43aeb20daf18c8ae832c0b0d924d0ec838d46d7e7943a0e4f1d2fbba47
ec05d055cbeebc1a0381264f1d1e29b19f737d1f5842e5a52f8a36a6433be7
fa853fc57f7c747274e4d56ebfe249d35b1eb38818b18f2247b7fad1d47cf755
84b7049c9aa13a311edf0d18663a82283e993bf8dbbb4b850b4b4812b99af7d
7bf51497be17b9f262780999b35175db42b16cbced1bda573ea85f55661c0d5c
4595ec1c1867db33213baa0b2a373ee64352b78ffdd008442ed55937717102c1
6147dd29e7b0ad02bb1351020d9fbc0d79b0091f948b43f8a7c4c132a03ba51
5fe21742a36e432bcdec329240ef8621439c7cc2360f80722d3495d967539cd
9fc897b5ea4a1a98ce4d1e5807639226f544b92285835744749cd6c284ab6
610f6e35ef8d0f7c35546d682eb085066fd7da531e42b521550a882314376
488516ee207255c3981e872ef1d607734e082187cf6800764b0992eb0e02226

```

```

7453e1b5f1ff09150633a05d0d3f3933b0305e36726dce6dff3fad90cf34e05c ./ftso/interface/IIFtso.sol
c5c4bdf3da71c72202233f06c0cfc4d484adb064add6b467a8e281812ba13ad9a ./ftso/interface/IIFtsoManagerV1.sol
795ed7e1637f23eacfla88da16aa2eb4785e65d2f9cac0c4d89a6b3f651fa9bf ./token/mock/VPTokenMock.sol
f9a26c45723f7222eb7fb92c03559b23f20393559b0a3a4be2bc552f189757a ./token/mock/DummyVPToken.sol
dce2c78c18163d8540271e4a4b74cafe680e80f8ac853442a81d847b7b5099c5 ./token/mock/TransferToWhatMock.sol
4c94b4f56769094663535fd042148691e5736ceaea8e87fcd59bd2f1cb5b919c ./token/mock/FlashLoanMock.sol
841498726cdf0115b452fcb8f35a657a6ee2d21a1e775a88ee7ea7f733b9b33ff ./token/mock/CheckPointsByAddressMock.sol
cea31e015368633710f9857b298c7f1a6ecfee141c2a30c4cd5cdf2d8e98e532 ./token/mock/PercentageDelegationMock.sol
666fe341bf0b0a5f0e28d7d339581eb89adb117fc095f99bd66d7dc37b3a5940 ./token/mock/CheckPointHistoryMock.sol
9f9022ac627b70ba075e57bb1aa978bbcb3b4d07ceb9be7e2ac f075294a4e41 ./token/mock/VotePowerMock.sol
8a06b88a399e75a55a404fb8fbc970db8643f6b47a5ce4ab24a306dc1eb4d3c ./token/mock/ExplicitDelegationMock.sol
4246d9082408e18033c1d2d5d6776242878a4cdeca24c696dd7e787a8f7fa8ec ./token/mock/VPContractMock.sol
85476a8b8d03782f81331edbd5527513f70b2d887222f68339d66fd7729d9057 ./token/mock/DelegatableMock.sol
91bd6388216144771835c99cfff1c6974d35d36ee5d1fb5a017066ee835b6c86 ./token/mock/CheckPointableMock.sol
cdf463a2c26f7245f3678565cb1275a5ffef6d633fa329e19c1e84e6af333d853 ./token/implementation/CheckPointable.sol
04a3bc74d57bbf9ab84cabfa308b8381db225db7a210a579f01319e6a3f89fff ./token/implementation/VPToken.sol
5003b4781ad9ad55718dbb6607ec6691078a8994779c15cf507fec52565e4a7e ./token/implementation/Delegatable.sol
aff920a71e29771dab9fedbd5614b303a21aa81b9bd2fe9a2fd0c3e3ca3426b ./token/implementation/WNat.sol
6b18c3c39097c659979825df30f912e80f4e9bd518e0b8b149d0ed8b4836c1dc ./token/implementation/CleanupBlockNumberManager.sol
23f612bdabff273dbff98deadcea84fe9fba7e3f2ab31c6804484f2a6904200c ./token/implementation/VPContract.sol
53aa232b7f50bb17d650419207735c18c7e90bdddedef074bc80e240c1442d01 ./token/lib/DelegationHistory.sol
796d9b62af94c15d89b94676d4514396e8a8c165c9e870e09996b3a0488cd63 ./token/lib/CheckPointHistoryCache.sol
7c7858e64134e57a69b4654e5e04153f448d346e246b84923d4339c4fb576751f ./token/lib/ExplicitDelegation.sol
82da0011307842f905d1b3809457ad58a0e869b7c0590a2edab9f6011e1f3c35 ./token/lib/PercentageDelegation.sol
ed07ecbf063aa6b852b4e7baf68d7a02485144db31ed37765cdab9b9e2396bf ./token/lib/VotePowerCache.sol
8ec06551b24efb1a3aef75ec0af76eca1f2845bad71eeafadd71ed514c0ab116 ./token/lib/VotePower.sol
8b964e7efa5b4db469251fb8b455f65964ed187f82707ff7eadc089ed41f8e6 ./token/lib/CheckPointsByAddress.sol
10f17a1c451bc43f55ba0a8a81a9a213cd1df6ec2bb7200cc21d13f422ed34dd ./token/lib/CheckPointHistory.sol
63856382a0837b1e219b5f5e49a402a90877d1e4b9d3469ade5add2613fd4b8a ./token/interface/IICleanable.sol
ad7bd71fb0f01dea138ff1fb0213100248873450301a2ad674285ba8cf56b05d ./token/interface/IIGovernanceVotePower.sol
c3f1fdbd8e3b1c298552586f4c07fc1bdd28c45d1d30dd765c3c0443afab3ca6 ./token/interface/IIVPContract.sol
37d9c059841c5c00853b6407cbb2cd15a8d5e144e7a0ecc039c7f818c0ccd94 ./token/interface/IIVPToken.sol

```

4. Summary of Finding

Id	Title	Total Risk	Fixed
FLR-6	Reward burner can be reconfigured to send funds to arbitrary account	Info	✓
FLR-7	priceDeviationThresholdBIPS can't adapt to market conditions	Info	✓
FLR-8	totalSelfDestructWithdrawnWei variable is never updated	Info	✓
FLR-9	Inflation annum tests not passing	Info	✓

5. Detailed Findings

FLR-6 Reward burner can be reconfigured to send funds to arbitrary account

Total Risk	Impact	Location
Info	-	UnearnedRewardBurner.sol FtsoRewardManager.sol Supply.sol
Fixed	Likelihood	
✓	-	

Description

The UnearnedRewardBurner contract is intended to burn funds that are transferred to it. This is performed via the `selfdestruct` mechanism. However, the destination address passed to `selfdestruct` is configurable, and as a result the funds could end up not being effectively burned but being sent to an address controlled by Governance.

```
function _burnUnearnedRewards() internal {
    // Are there any rewards to burn?
    uint256 rewardsToBurnWei = totalUnearnedWei.add(totalExpiredWei).sub(totalBurnedWei);

    if (rewardsToBurnWei > 0) {
        // Calculate max rewards that can be burned
        uint256 maxToBurnWei = address(this).balance.mulDiv(MAX_BURNABLE_PCT, 100);

        uint256 toBurnWei = 0;
        // Calculate what we will burn
        if (rewardsToBurnWei > maxToBurnWei) {
            toBurnWei = maxToBurnWei;
        } else {
            toBurnWei = rewardsToBurnWei;
        }

        // Any to burn?
        if (toBurnWei > 0) {
            // Get the burn address; make it payable
            address payable burnAddress = payable(supply.burnAddress());

            // Accumulate what we are about to burn
            totalBurnedWei = totalBurnedWei.add(toBurnWei);
        }
    }
}
```

```

// Update lastBalance before transfer, to avoid reentrancy warning
lastBalance = lastBalance.sub(toBurnWei);

// Burn baby burn
UnearnedRewardBurner unearnedRewardBurner = new
    UnearnedRewardBurner(burnAddress);
//slither-disable-next-line arbitrary-send
address(unearnedRewardBurner).transfer(toBurnWei);
unearnedRewardBurner.die();

// Emit event to signal what we did
emit RewardsBurned(toBurnWei);
    }
}
}

```

The target `burnAddress` is obtained from the `Supply` contract, which allows Governance to modify the address at will:

```

/**
 * @notice Change burn address
 * @param _burnAddress          New burn address
 * @dev Updates burn value for current address, changes to new address and updates again
 */
function changeBurnAddress(address _burnAddress) external onlyGovernance {
    _updateCirculatingSupply();
    burnAddressBalance = 0;
    burnAddress = _burnAddress;
    _updateBurnAddressAmount();
}

```

Recommendation

Consider removing the ability to change the destination address of the funds sent to the reward burner contract. Alternatively, document it is possible for Governance to configure the protocol to send the funds to any account instead of burning them as suggested by the contract name.

Status

This issue was addressed in commit [f66b9a5af48ce46e4d2fbbe4e80a29cea93b20fc](#).

FLR-7

priceDeviationThresholdBIPS can't adapt to market conditions

Total Risk	Impact	Location
Info	-	Ftso.sol
Fixed	Likelihood	
✓	-	

Description

The `priceDeviationThresholdBIPS` storage variable is immutable. As a consequence, it would require redeploying the contract to modify this parameter in order to allow the platform to react to potentially quick market changes.

```
// storage
uint256 public immutable priceDeviationThresholdBIPS;
// threshold for price deviation between consecutive epochs
```

The variable is set by the contract constructor and is utilized by the protection mechanism that prevents the oracle price from moving too fast and triggers the fallback mechanism.

For that reason it would be ideal if the variable can be easily modified in order to adapt to a changing market context without the cost of redeploying the contract.


Recommendation

Consider making this variable configurable by governance.

Status

Flare confirmed that the `priceDeviationThresholdBIPS` value is updated by re-deploying the FTSO contract and replacing it.

FLR-8
totalSelfDestructWithdrawnWei variable is never updated

Total Risk	Impact	Location
Info	-	Inflation.sol
Fixed	Likelihood	
	-	

Description

The `totalSelfDestructWithdrawnWei` storage variable is never updated and remains uninitialized. However, it is used by two functions, for example:

```
function getExpectedBalance() private view returns(uint256 _balanceExpectedWei) {
    return inflationAnnums.totalInflationTopupReceivedWei
        .sub(inflationAnnums.totalInflationTopupWithdrawnWei)
        .add(totalSelfDestructReceivedWei)
        .sub(totalSelfDestructWithdrawnWei);
}
```

This issue does not currently represent a risk as the value is always 0.

This was observed in the `coinspect_audit2_branch1`, the variable does not exist in `coinspect_audit2_branch2` branch.


Recommendation

Remove the unused variable.

Status

The variable `totalSelfDestructWithdrawnWei` is not present in commit [42ff03c74f10f816e606d02f2fb6b741a3400195](https://github.com/coinspect/coinspect/commit/42ff03c74f10f816e606d02f2fb6b741a3400195).

FLR-9
Inflation annum tests not passing

Total Risk	Impact	Location
Info	-	Inflation.sol
Fixed	Likelihood	
	-	

Description

Coinspect observed two tests from the Inflation contract unit tests were not passing in the auditors test boxes. The tests yielded a different result when being executed in an isolated manner than when run as a part of the full test suite.

```

1) Contract: Inflation.sol; test/unit/inflation/implementation/Inflation.ts;
Inflation unit tests
  init
    Should initialize the annum:
    AssertionError: expected '366' to equal 365
      at Context.<anonymous>
(test/unit/inflation/implementation/Inflation.ts:152:14)
      at runMicrotasks (<anonymous>)
      at processTicksAndRejections (node:internal/process/task_queues:96:5)
2) Contract: Inflation.sol; test/unit/inflation/implementation/Inflation.ts;
Inflation unit tests
  recognize
    Should recognize new annum when year rolls over:
    AssertionError: expected 100000 to equal 200000
    + expected - actual
    -100000
    +200000

      at Context.<anonymous>
(test/unit/inflation/implementation/Inflation.ts:186:14)
      at runMicrotasks (<anonymous>)
      at processTicksAndRejections (node:internal/process/task_queues:96:5)

```

Upon consultation, the Flare team stated the tests were passing when executed from their automated build process.

These tests will be reviewed by the Flare team.

Recommendation

Further investigate the tests to determine the failure root cause and fix them if necessary.

Status

This issue was addressed in commit [81ab059a5189a5cec6b0babae7d9c31df3b4cbfa](#).

5. Disclaimer

The information presented in this document is provided “as is” and without warranty. Security Audits are a “point in time” analysis, and as such, it’s possible that something in scope may have changed since the tasks reflected in this report were executed. This report shouldn’t be considered a perfect representation of the risks threatening the analyzed systems and/or applications in scope.

6. About Coinspect

Coinspect is a boutique-style security consulting firm focused on Blockchain, Cryptocurrencies and Web3 technologies. Founded in 2014 by a team of information security professionals with currently +20 years of experience providing valuable and outstanding results to cutting-edge technology companies worldwide.

- Team of expert auditors (with +6y of experience in crypto / web3)
- Tailored service delivery to each client and their specific targets(s).
- Own manual security audit methodology (not a product/platform centric).
- Contributing to the community by publishing papers, techniques, and tools.

We have contributed to secure key technologies and services of the cryptocurrency ecosystem by auditing, reporting vulnerabilities, and proposing improvements for new blockchain (L1/L2) implementations, smart contracts, dApps, mobile apps, hardware wallets, compilers, developer tools, DeFi protocols, and exchanges.

Delivering True Value to our Customers

- Lower Cost of Vulnerability Remediation; by delivering a detailed vulnerability report and included mitigation recheck services, our team will interact closely with Customers’ team and educate them on findings and recommendations.
- Increase Confidence and Expand Adoption; by providing a reliable technology from a security standpoint the users will increase the circulating value in the platforms.
- Lessen Business Impact of Incidents; our state-of-the-art manual security assessment ensures that any potential threats are identified and addressed quickly and effectively, minimizing the business impact of incidents.

For more information, please visit our website <https://www.coinspect.com//>