Staker Implementation and TGE Payload

Security Review

Cantina Managed review by:

Desmond Ho, Lead Security Researcher

Arno, Associate Security Researcher

January 22, 2026

Contents

1 Introduction

1.1 About Cantina

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

1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity level	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

1.3.1 Severity Classification

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

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

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

2 Security Review Summary

Aztec Labs was founded in 2017, and has a team of +50 leading zero-knowledge cryptographers, engineers, and business experts. Aztec Labs is developing its namesake products: AZTEC (a privacy-first L2 on Ethereum) and NOIR (the universal ZK language).

From Jan 15th to Jan 16th the Cantina team conducted a review of ignition-contracts on commit hash 3d54af02. The team identified a total of **2** issues:

Issues Found

Severity	**Count**	**Fixed**	**Acknowledged**
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	0	0	0
Gas Optimizations	0	0	0
Informational	2	2	0
Total	**2**	**2**	**0**

2.1 Scope

The security review had the following components in scope for ignition-contracts on commit hash 3d54af02:

```
src
├── staking
│   ├── ATPNonWithdrawableStaker.sol
│   └── ATPWithdrawableStaker.sol
└── tge
    ├── ATPWithdrawableAndClaimableStakerV2.sol
    └── TGEPayload.sol
```

3 Findings

3.1 Informational

3.1.1 Edge case test

Severity: Informational

Context: ATPWithdrawableAndClaimableStakerV2.sol#L38-L43

Description/Recommendation: The no-op case was untested. A test case was written to ensure its intended behaviour.

```diff
  diff --git a/test/tge/Delegation.t.sol b/test/tge/Delegation.t.sol
  index 284f9fb..1588126 100644
-  -- a/test/tge/Delegation.t.sol
+  ++ b/test/tge/Delegation.t.sol
  @@ -4,12 +4,42 @@ pragma solidity ^0.8.13;
   import {IATPWithdrawableAndClaimableStaker} from
    ↪  "src/staking/interfaces/IATPWithdrawableAndClaimableStaker.sol";
   import {Governance} from "@aztec/governance/Governance.sol";
   import {NCATP} from "src/token-vaults/atps/noclaim/NCATP.sol";
-
+  import {BN254Lib, G1Point, G2Point} from "@aztec/shared/libraries/BN254Lib.sol";
   import {Base} from "./Base.sol";
+  import {GSECore, GSE} from "@aztec/governance/GSE.sol";
+  import {ATPNonWithdrawableStaker} from
↪   "src/tge/ATPWithdrawableAndClaimableStakerV2.sol";
+  import {Registry, IHaveVersion} from "@aztec/governance/Registry.sol";
+
+  contract MinimalRollup is IHaveVersion {
+      function getVersion() external view returns (uint256) {
+          return 1;
+      }
+
+      function getGSE() external view returns (GSE) {
+          return GSE(address(0xa92ecFD0E70c9cd5E5cd76c50Af0F7Da93567a4f));
+      }
+  }

   contract DelegationTest is Base {
       NCATP public ATP = NCATP(0xE1ea32a54F4FB323dBbE760384617CAa7aa0f331);
       address public ATTESTER = 0x0Ce7B6316E7dA7d02f6f98001296bb7E77aaDAE1;
+      address public NEW_ATTESTER = makeAddr("NEW_ATTESTER");
+
+      /// @notice Generate G1 public key from private key
+      /// @param sk The private key (scalar)
+      /// @return The G1 public key point
+      function generateG1Key(uint256 sk) internal view returns (G1Point memory) {
+          return BN254Lib.g1Mul(BN254Lib.g1Generator(), sk);
+      }
+
+      /// @notice Generate proof of possession signature from private key
+      /// @param sk The private key (scalar)
+      /// @return The proof of possession signature (G1 point)
+      function generateProofOfPossession(uint256 sk) internal view returns (G1Point
↪   memory) {
+          G1Point memory pk1 = generateG1Key(sk);
+          G1Point memory pk1DigestPoint = BN254Lib.g1ToDigestPoint(pk1);
+          return BN254Lib.g1Mul(pk1DigestPoint, sk);
+      }

       function test_delegateFollowingToSelf() public {
           IATPWithdrawableAndClaimableStaker staker =
            ↪  IATPWithdrawableAndClaimableStaker(address(ATP.getStaker()));
  @@ -39,4 +69,90 @@ contract DelegationTest is Base {
           // Ensure that it increased! It is now working, wuhu
```

```
            assertGt(GSE.getVotingPower(beneficiary), beneficiaryPowerBefore);
        }
+
+       function test_delegateExitedAttester() public {
+           MinimalRollup rollup = new MinimalRollup();
+           uint256 sk = 125269879;
+
+           // Generate G1 public key on-chain
+           G1Point memory g1Key = generateG1Key(sk);
+
+           // G2 public key must be computed off-chain (no g2Mul in BN254Lib)
+           // You can use FFI or a library like gnark-crypto to compute:
+           // g2Key = g2Generator * sk
+           // For now, using precomputed values (you can verify these match by checking
+           // that proofOfPossession verification passes)
+           G2Point memory g2Key = G2Point({
+               x0:
 ↪  17931071651819835067098563222910421513876328033572114834306979690881549564414,
+               x1:
 ↪  12000187580290590047264785709963395816646295176893602234201956783324175839805,
+               y0:
 ↪  9611549517545166944736557219282359806761534888544046901025233666228290030286,
+               y1:
 ↪  3847186948811352011829434621581350901968531448585779990319356482934947911409
+           });
+
+           // Generate proof of possession on-chain
+           G1Point memory proofOfPossession = generateProofOfPossession(sk);
+
+           IATPWithdrawableAndClaimableStaker staker =
 ↪  IATPWithdrawableAndClaimableStaker(address(ATP.getStaker()));
+
+           address operator = ATP.getOperator();
+           address beneficiary = ATP.getBeneficiary();
+
+           // 1. Execute the proposal and upgrade to V2 staker
+           proposeAndExecuteProposal();
+           vm.prank(beneficiary);
+           ATP.upgradeStaker(STAKER_VERSION);
+
+           // 2. Add the rollup, mint activation threshold tokens and call deposit() with
 ↪  moveWithLatestRollup as false
+           address gseOwner = GSECore(address(GSE)).owner();
+           vm.prank(gseOwner);
+           GSE.addRollup(address(rollup));
+
+           // add rollup to registry
+           Registry registry =
 ↪  Registry(address(ATPNonWithdrawableStaker(address(staker)).ROLLUP_REGISTRY()));
+           vm.prank(registry.owner());
+           registry.addRollup(IHaveVersion(address(rollup)));
+
+           uint256 activationThreshold = GSECore(address(GSE)).ACTIVATION_THRESHOLD();
+           deal(address(AZTEC_TOKEN), address(rollup), activationThreshold);
+           vm.startPrank(address(rollup));
+           AZTEC_TOKEN.approve(address(GSE), activationThreshold);
+           GSE.deposit(
+               NEW_ATTESTER,
+               address(staker),
+               g1Key,
+               g2Key,
+               proofOfPossession,
+               false
+           );
+           vm.stopPrank();
+
+           // Verify attester is registered
+           assertTrue(GSE.isRegistered(address(rollup), NEW_ATTESTER), "Attester should
↪   be registered initially");
+
+           uint256 beneficiaryPower = GSE.getVotingPower(beneficiary);
+
+           // 3. Staker delegates to the new attester, voting power should have increased
+           vm.prank(operator);
+           staker.delegate(1, NEW_ATTESTER, beneficiary);
+
+           assertGt(GSE.getVotingPower(beneficiary), beneficiaryPower);
+
+           // 4. Attester exits
+           vm.prank(address(rollup));
+           GSE.withdraw(NEW_ATTESTER, activationThreshold);
+           assertFalse(GSE.isRegistered(address(rollup), NEW_ATTESTER), "Attester should
↪   have been de-registered");
+
+           // beneficiary power should be 0 after attester exit
+           assertEq(GSE.getVotingPower(beneficiary), 0);
+
+           // 5. Staker tries to delegate, should be a no-op
+           vm.prank(operator);
+           staker.delegate(1, NEW_ATTESTER, beneficiary);
+           assertEq(GSE.getVotingPower(beneficiary), 0);
+
+           // 6. Delegate to previous version that fallbacks to bonus version, voting
↪   power should have increased
+           vm.prank(operator);
+           staker.delegate(0, ATTESTER, beneficiary);
+           assertGt(GSE.getVotingPower(beneficiary), 0);
+       }
    }
```

Aztec Labs: Fixed in commit 448103a2.

Cantina Managed: Fix verified. A variant to this test scenario was added in the PR.

3.1.2 Comment Typo

Severity: Informational

Context: ATPNonWithdrawableStaker.sol#L158-L178

Description/Recommendation: There is a typo - "`Goverance`" Instead of "`Governance`":

```
- Goverance
+ Governance
```

Aztec Labs: Fixed in commit 448103a2.

Cantina Managed: Fix verified.