SMCAUDIT

SMART CONTRACT
SECURITY AUDIT



AUDIT REPORT

OCICAT Ecosystem Security Review

PROJECT	OCICAT
CHAIN	BNB Smart Chain (chain id 56)
CONTRACTS	OCICAT token, staking farm, NFT, DXLock LP lock (see Scope)
VERSION	Verified live on-chain bytecode · Solidity 0.8.20+ (token 0.8.21, NFT 0.8.20, farm 0.8.24)
REPORT ID	SMC-OCICAT-202610-01
DATE	10 October 2026
PREPARED BY	SMC Audits
STATUS	Final

0 0

00 Confidentiality notice

This report is prepared by SMC Audits for the named project and contains confidential information, including identified security findings.

It should not be shared publicly until the project has remediated the issues described, and only with SMC Audits' agreement. Publication as part of public verification follows the process agreed in the engagement.

SMC AUDITS

01

01 Contents

02	Executive summary
03	Project overview
04	Scope
05	Methodology
06	Severity classification
07	Findings summary
08	Detailed findings
09	Remediation status
10	Conclusion
11	Disclaimer

SMC AUDITS

02

02 Executive summary

SMC Audits reviewed the OCICAT ecosystem smart contracts on BNB Smart Chain at their deployed addresses. The review covered the whole ecosystem: the OCICAT (V2) token, the deprecated V1 token, the WBNB/OCICAT liquidity-pool position and its DXLock time-lock, the MultiStakeRewards staking farm, and the OcicatNFT governance-weight NFT.

The review identified 20 findings: 0 critical, 2 high, 3 medium, 8 low, and 7 informational. The most significant issues are in the staking farm. A decimal-mismatch defect over-issues rewards by a factor of about 1e12 and leaves the farm structurally insolvent (SMC-H-01), and the owner can withdraw the entire reward reserve at any time (SMC-H-02). The token and NFT are simpler and carry low-severity hygiene and centralization issues only.

Overall, the token and NFT are free of the most dangerous classes. There is no arbitrary mint, no proxy or upgrade path, no exploitable reentrancy, no oracle or flash-loan surface, and user principal, holder balances and staked LP, cannot be stolen by an attacker. The material risk is concentrated in the farm's reward accounting and in broad, EOA-held owner privileges across the ecosystem. 76.3% of the LP is time-locked until 29 April 2028 and cannot be withdrawn early.

0	2	3	8	7	20
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

As of this report, 0 of 20 findings have been resolved. No contract in scope is a proxy, so the bytecode cannot be patched in place. The recommendations are advisory for the live deployment: they apply to the owner, token holders, farm stakers, integrators, and any future redeployment. Any residual risk is described in the conclusion.

SMC AUDITS

03

03 Project overview

OCICAT is a fee-on-transfer memecoin on BNB Smart Chain with an NFT collection and an LP-staking farm that awards "DAO power." This audit covers the token, the farm, the NFT, and the DXLock LP lock, plus the deprecated V1 token. No contract in scope is upgradeable and none reads a price oracle.

System at a glance

COMPONENT	ROLE

OcicatToken (V2)	Core BEP-20: balances and transfers, with a fixed 3% admin fee plus 2% burn on non-whitelisted sends. There is no setter for the fee rates (effectively immutable), and no mint, pause, or blacklist.
MultiStakeRewards (farm)	Stakes WBNB/OCICAT LP and emits OCICAT rewards, with cooldown and emergency exits and a DAO-power tally. Rewards are computed from the staked amount scaled by a fixed divisor.
OcicatNFT	ERC-721 that also implements IDAOPower (governance weight = balance x nftPower). Public mint plus whitelist claim. The NFT owner is immutable.
DXLock LP locker	Third-party time-lock holding 76.3% of the LP until 29 April 2028. Both owner withdraw paths revert before expiry.
Admin / ownership	Single externally-owned keys hold the token admin and the farm owner roles. The farm owner holds broad privileges: an uncapped reward-token withdrawal, an uncapped cooldown period, and an arbitrary DAO-power setter.

Key mechanisms to note

- The token charges a fixed 3% fee plus 2% burn on transfers from non-whitelisted senders, with no rate setter, no mint, no pause, and no blacklist.
- The farm computes rewards from the staked amount scaled by a fixed divisor. The staking token (the WBNB/OCICAT LP) is 18-decimal and the reward token (OCICAT) is 6-decimal, and the scaling does not reconcile the two. This is the SMC-H-01 defect and it makes the farm insolvent by design.
- The farm owner can withdraw the entire reward reserve at any time, can set an unbounded cooldown that forces penalized exits, and can assign arbitrary DAO power.
- The NFT's daoPower is read live from a freely transferable balance. No on-chain contract currently consumes it, so it is advisory today.

SMC AUDITS

04

04 Scope

ITEM	DETAIL
Project	OCICAT
Chain	BNB Smart Chain (chain id 56)
Language	Solidity
Commit / version	Verified live on-chain bytecode; confirm the exact in-scope source for the final. No contract is a proxy.

Compiler	Solidity 0.8.20 or later (token 0.8.21, NFT 0.8.20, farm verified at 0.8.24), optimizer per verified metadata.

Addresses

CONTRACT	ADDRESS
OcicatToken (V2)	0xE53D384Cf33294C1882227ae4f90D64cF2a5dB70
MultiStakeRewards (farm)	0xD84A6CBe328D761335A4a46f849F93788769d825
OcicatNFT	0x90bdea0ddb6160faf6115dc35317c05cc911be22
WBNB/OCICAT LP pair	0x1df65d3a75AeCd000A9c17c97E99993aF01DbcD1
DXLock LP locker	0x5e03329fec9e9e3d296171c3516f21887bc0f7fd
OciCat V1 (deprecated)	0x37fe635d1e25b2f7276c1b9dbbcc7b087f80c050

Files reviewed

FILE	LINES	NOTES
OcicatToken.sol	~148	Core BEP-20, fee-on-transfer, burn.
MultiStakeRewards.sol	~295	LP staking farm, reward emission, cooldown and emergency exits.
OcicatNFT.sol	~363	ERC-721 plus IDAOPower, public mint plus whitelist claim.
PersonalLPLocker.sol (DXLock)	—	Third-party lock, reviewed for lock integrity only.

Out of scope

The DXLock platform internals (reviewed only to confirm the LP lock cannot be released early), the off-chain frontends (reviewed passively for secret leakage only), deployment and key management, and market or economic conditions. Proofs were run against the live bytecode on a BNB-chain fork and against local deployments of the verified source. No live mainnet transaction was made.

05

05 Methodology

Every SMC Audits review follows the same process, so the result means the same thing on every engagement.

1. Threat modeling. Understand what the contracts do, where value moves, and who holds privilege.

2. Manual review. A line-by-line reading of the code by the audit team. This is the core of the work and where most issues are found.
3. Tooling. Static analysis and security tooling support the manual review, they do not replace it.
4. Proof of concept. Every material finding is confirmed with a working proof of concept rather than asserted.
5. Reporting. Findings are written up with severity, reproduction, and a clear recommendation.
6. Re-audit. Fixes are verified by the team to confirm the issue is closed without introducing a new one.

Review checklist

Access control, signer and owner checks; privilege and centralization risk; arithmetic, overflow, and precision; reentrancy and external call safety; token and supply logic, mint and burn; fee, tax, and reflection mechanics; approvals and allowance handling; oracle and price handling; upgradeability and proxy risk; pause, blacklist, and emergency switches; front-running and MEV exposure; denial of service and gas griefing.

06

06 Severity classification

SEVERITY	MEANING
CRITICAL	Direct loss of funds or full compromise, realistic to exploit
HIGH	Serious impact, exploitable under conditions a project would actually meet
MEDIUM	Meaningful risk, limited by conditions or the size of impact
LOW	Minor risk, hard to exploit or low in impact
INFO	Best practice and code quality notes worth addressing over time

07

07 Findings summary

ID	TITLE	SEVERITY	STATUS

SMC AUDITS

SMC-H-01	Farm reward over-issuance (~1e12x) drains the reserve and leaves the farm insolvent	HIGH	Open
SMC-H-02	Farm owner can drain the entire reward reserve at any time	HIGH	Open
SMC-M-01	Owner can force up to 50% penalized exits via an unbounded cooldown	MEDIUM	Open
SMC-M-02	Staking during an active cooldown permanently strands the deposit	MEDIUM	Open
SMC-M-03	Reward insolvency bricks the reward leg of both exit paths	MEDIUM	Open
SMC-L-01	unstake() forfeits all accrued rewards	LOW	Open
SMC-L-02	Token fee and burn transfers emit no Transfer event	LOW	Open
SMC-L-03	Token transferOwnership has no zero-address guard	LOW	Open
SMC-L-04	Unbounded userStakes[] growth (self-inflicted gas DoS)	LOW	Open
SMC-L-05	NFT daoPower is liquid and owner-inflatable	LOW	Open
SMC-L-06	NFT batchTransfer / batchTransferTo are non-functional	LOW	Open
SMC-L-07	Buy/sell tax asymmetry, non-FoT router sells revert	LOW	Open
SMC-L-08	Farm emergencyUnstake does not decrement totalDaoPower	LOW	Open
SMC-I-01	NFT owner is immutable (no rotation to multisig)	INFO	Open
SMC-I-02	NFT owner can free-mint remaining supply and rescale power	INFO	Open
SMC-I-03	Admin/owner roles are single EOAs, not multisigs	INFO	Open
SMC-I-04	Farm assignDaoPower and retroactive setRewardSettings	INFO	Open
SMC-I-05	NFT shared mint/claim supply, full-allocation claim	INFO	Open
SMC-I-06	Admin setters emit no events	INFO	Open
SMC-I-07	recoverERC20 ignores transfer return, dust fee rounding	INFO	Open

SMC AUDITS

0	2	3	8	7	20
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

08

08 Detailed findings

PoC note: fork tests select a BNB-chain fork and run against the live bytecode. Local tests deploy the verified source. No live mainnet transaction was made. The full tests are provided in the engagement materials.

SMC AUDITS

SMC-H-01 HIGH Open

Farm reward over-issuance (~1e12x) drains the reserve and leaves the farm insolvent

CATEGORY

Token and reward logic / arithmetic

LOCATION

MultiStakeRewards.sol, getPendingRewards(address) (line 152)

DESCRIPTION

Rewards are computed as `reward = stakes[i].amount * REWARD_PERCENTAGE * intervals / 1e6`. The staking token (the WBNB/OCICAT LP) has 18 decimals and the reward token (OCICAT) has 6 decimals. The `/1e6` encodes the intended per-interval rate, but it is applied to an 18-decimal quantity and paid out in a 6-decimal token. The missing decimal normalization, a further `/1e12`, inflates every reward by about 1e12. By raw mechanism this is Critical-class, permissionless and an atomic drain of a pool. It is rated High here because the drained asset is the team-funded OCICAT reward reserve, not user principal or holder balances.

IMPACT

Staking 1 LP token for 1 hour entitles the staker to 60,000,000 OCICAT (1,000,000 OCICAT per LP per minute). Against the live reserve of about 1,899,975,374 OCICAT, a single 1-LP staker's accrual surpasses the entire reserve after about 31.7 hours, and at the roughly 5.6 LP currently staked, combined accrual surpasses it in about 5.7 hours. Over the 365-day interval cap, one LP can claim up to about 5.26 x 10^11 OCICAT, roughly 277 times the whole reserve. The first claimant empties the reserve and every later claimant reverts on an insufficient balance. The farm is already insolvent for its promised emissions. The exploit is permissionless, and it is bounded to the reward reserve and cannot reach holder balances or staked LP principal.

PROOF OF CONCEPT

On a BNB-chain fork, 1 LP staked for 1 hour returns getPendingRewards = 60,000,000 OCICAT, and one claimRewards transfers that amount out of the reserve. The over-issuance factor is pinned exactly: for 1 LP over 1 hour the formula yields 6 x 10^13 raw units (60,000,000 OCICAT), whereas the decimal-correct value is 60 raw units, a factor of 1e12. A second fork test shows that with a small reserve a single staker's accrual exceeds the reserve, so the claim and both exit reward legs revert.

RECOMMENDATION

Normalize across token decimals exactly once, multiply by 10**rewardDecimals and divide by 10**stakeDecimals (here, add the missing `/1e12`). Better, decouple emission from stake magnitude with a funded, rate-limited reward-per-token accumulator, and require reward <= availableReserve before each payout so the farm can never promise more than it holds.

SMC-H-02 HIGH Open

Farm owner can drain the entire reward reserve at any time

CATEGORY

Access control / centralization

LOCATION

MultiStakeRewards.sol, emergencyWithdrawTokens(address,uint256) (lines 284-287)

DESCRIPTION

`emergencyWithdrawTokens(to, amount)` is owner-only but has no cap and no carve-out for rewards already accrued or owed. The owner can transfer the full OCICAT reward balance to any address at any time. The function handles the reward token only and cannot move the staked LP, so principal is not reachable this way.

IMPACT

Staker rewards are entirely discretionary. The owner can empty the reward pool whenever they choose, leaving stakers unable to claim. Verified against source, the function carries only `onlyOwner` and `nonReentrant`, with no amount bound. User LP principal is not reachable through this path.

PROOF OF CONCEPT

A fork test funds the farm with an arbitrary reward balance, pranks the owner, and calls emergencyWithdrawTokens(sink, fullBalance). The farm's reward balance ends at 0 and the full amount lands in an owner-chosen address, confirming an unconditional, total drain of the reserve.

RECOMMENDATION

Place ownership behind a multisig and timelock. Escrow accrued rewards so they cannot be withdrawn out from under stakers. If an emergency withdrawal must exist, cap it to the non-accrued surplus.

SMC AUDITS

SMC-M-01 MEDIUM Open

Owner can force up to 50% penalized exits via an unbounded cooldown

CATEGORY

Centralization / denial of service

LOCATION

MultiStakeRewards.sol, setCooldownPeriod(uint256) (line 254), setEmergencyFee(uint256) (line 260)

DESCRIPTION

setCooldownPeriod enforces only a lower bound (>= 1 minute) with no upper bound, verified against source. A malicious or compromised owner can set a very long cooldown so that withdrawAfterCooldown is unreachable in practice, forcing stakers onto emergencyUnstake, which applies a penalty of up to 50% of their staked LP (the emergencyFee cap) paid to the owner.

IMPACT

Up to 50% of a staker's LP is forced to the owner as penalty, about 11.6% of the total pool at the current staked share. This affects farm stakers only and does not reach holder balances. The owner must act maliciously, the staker must choose to exit, and the penalty is capped at 50% with the remainder returned, hence Medium.

PROOF OF CONCEPT

A fork test stakes 100 LP, then the owner sets a 10-year cooldown and a 50% emergency fee. withdrawAfterCooldown reverts for the full window, and the forced emergencyUnstake returns 50 LP to the user and 50 LP to the owner.

RECOMMENDATION

Cap cooldownPeriod (for example, 30 days or less). Consider sending the emergency penalty to a burn address or treasury rather than the owner, and place ownership behind a multisig and timelock.

SMC AUDITS

SMC-M-02 MEDIUM Open

Staking during an active cooldown permanently strands the deposit

CATEGORY

Accounting / fund loss

LOCATION

MultiStakeRewards.sol, unstake() (lines 179-192), stake(uint256) (line 120)

DESCRIPTION

unstake() snapshots cooldowns.amount = totalUserStaked and deletes userStakes, but it does not zero totalUserStaked, and stake() has no guard against staking while a cooldown is active. The sequence: stake 100, unstake(), then stake 50 during cooldown (totalUserStaked becomes 150), then withdrawAfterCooldown() pays the snapshot of 100, zeroes totalUserStaked, and deletes userStakes, discarding the extra 50.

IMPACT

The user deposits 150 LP, recovers 100, and 50 LP is permanently stranded with no user claim and no owner rescue (the owner cannot move the staking token). This is a permissionless user-fund loss via a plausible self-sequence, hence Medium.

PROOF OF CONCEPT

A fork test runs the stake, unstake, stake-during-cooldown, withdrawAfterCooldown sequence with an oversized reserve so the reward leg clears and the LP loss is isolated. The user recovers only the 100 LP cooldown snapshot and totalUserStaked ends at 0, with the extra 50 LP stranded.

RECOMMENDATION

Block stake() while a cooldown is active, or fold the new deposit into the cooldown. On unstake(), set totalStaked -= totalUserStaked and totalUserStaked = 0, so that sum(totalUserStaked) == totalStaked holds.

SMC AUDITS

SMC-M-03 MEDIUM Open

Reward insolvency bricks the reward leg of both exit paths

CATEGORY

Accounting / denial of service

LOCATION

MultiStakeRewards.sol, withdrawAfterCooldown() (line 211), emergencyUnstake() (line 235)

DESCRIPTION

Both exits require rewardToken.transfer(user, rewards) in the same transaction as the LP return. Because SMC-H-01 over-issues rewards, accrued rewards quickly exceed any realistic reserve, so the reward transfer reverts and takes the whole exit with it.

IMPACT

Once rewards over-accrue, both the cooldown exit and the emergency exit revert on the reward leg. The only escape becomes unstake(), which forfeits all accrued rewards (SMC-L-01). Exit reliability is coupled to reward solvency.

PROOF OF CONCEPT

With a small reserve, a single 1-LP staker's accrued reward exceeds the reserve after one hour, so claimRewards and the identical reward leg in the exits revert on an insufficient balance.

RECOMMENDATION

Pay the LP principal unconditionally. Pay rewards separately and best-effort, or bounded to the available reserve, so a reward shortfall can never trap a user's principal.

SMC AUDITS

SMC-L-01 LOW Open

unstake() forfeits all accrued rewards

CATEGORY

Reward logic

LOCATION

MultiStakeRewards.sol, unstake() (lines 179-192)

DESCRIPTION

unstake() sets the cooldown and runs delete userStakes[msg.sender] without paying rewards. withdrawAfterCooldown then computes rewards over the now-empty array and pays 0. A user who does not call claimRewards() immediately before unstaking loses all accrued rewards.

IMPACT

Silent loss of accrued rewards for any user who unstakes without claiming first. Principal is unaffected, hence Low.

PROOF OF CONCEPT

A fork test accrues rewards, calls unstake(), and confirms getPendingRewards drops to 0, then withdrawAfterCooldown pays 0 rewards.

RECOMMENDATION

Snapshot and pay, or carry into the cooldown, pending rewards inside unstake().

SMC-L-02 LOW Open

Token fee and burn transfers emit no Transfer event

CATEGORY

Standards compliance

LOCATION

OcicatToken.sol, _transfer(address,address,uint256) (lines 99-109)

DESCRIPTION

The admin-fee credit and the burn change balances and supply but emit only custom FeePaid and Burn events, not the EIP-20 Transfer event. A taxed transfer emits exactly one Transfer carrying only the net amount, so explorers and indexers that track balances via Transfer under-see the sender's outflow by the fee plus burn.

IMPACT

Indexers under-count the sender's outflow. No balance or theft impact, hence Low.

PROOF OF CONCEPT

A taxed 500,000-unit transfer emits exactly one Transfer of the net 475,000. The fee (15,000) and burn (10,000) move with no Transfer event, confirmed on both a local deploy and the live contract.

RECOMMENDATION

Emit Transfer(sender, feeReceiver, feeAmount) and Transfer(sender, address(0), burnAmount).

SMC AUDITS

SMC-L-03 LOW Open

Token transferOwnership has no zero-address guard

CATEGORY

Access control

LOCATION

OcicatToken.sol, transferOwnership(address) (lines 144-147)

DESCRIPTION

Unlike setFeeReceiver and the constructor, transferOwnership sets admin with no != address(0) check, and it is single-step. A slip to the zero address permanently bricks every onlyAdmin function.

IMPACT

An accidental transfer to the zero address permanently disables admin control. Requires an owner mistake, hence Low.

PROOF OF CONCEPT

A local test sets admin to address(0) via transferOwnership, after which every onlyAdmin path, such as setFeeReceiver, reverts with "Caller is not admin".

RECOMMENDATION

Add require(newAdmin != address(0)), and prefer a two-step (pending/accept) ownership transfer.

SMC AUDITS

SMC-L-04 LOW Open

Unbounded userStakes[] growth (self-inflicted gas DoS)

CATEGORY

Denial of service

LOCATION

MultiStakeRewards.sol, stake(uint256) (line 120), reward loops in getPendingRewards and claimRewards

DESCRIPTION

Each stake() pushes a new array element, and the reward loops iterate the array. A user who stakes many times grows their own loop until their claims and exits run out of gas. The growth is keyed to the caller's own account (no cross-user impact), and unstake()'s delete clears it.

IMPACT

A user can only grief themselves, and the state is recoverable via unstake(). No cross-user impact, hence Low.

PROOF OF CONCEPT

A test stakes 300 times from one account and shows getPendingRewards gas growing linearly, while a victim's single-stake read stays at about 6.8k gas.

RECOMMENDATION

Aggregate per-user stake into a single accumulator rather than an array.

SMC AUDITS

SMC-L-05 LOW Open

NFT daoPower is liquid and owner-inflatable (latent)

CATEGORY

Governance

LOCATION

OcicatNFT.sol, daoPower(address) (lines 348-350), setNFTPower(uint256) (line 268)

DESCRIPTION

daoPower(user) = balanceOf(user) * nftPower, read live from a freely transferable NFT balance, and nftPower is owner-settable. Governance weight is fully liquid and the owner can rescale everyone's weight. This is latent today. No on-chain contract consumes daoPower (verified across the deployed set), so no funds or state are currently at risk.

IMPACT

If daoPower ever gates on-chain decisions, weight could be bought, moved atomically, or rescaled by the owner. No consumer exists today, hence Low.

PROOF OF CONCEPT

A test mints 10 NFTs and shows daoPower = 10 x nftPower instantly, then transfers the NFTs and shows the weight moves atomically to the buyer.

RECOMMENDATION

If daoPower will gate on-chain decisions, snapshot it (checkpoints) and bound nftPower. Otherwise, document that it is advisory and off-chain only.

SMC AUDITS

SMC-L-06 LOW Open

NFT batchTransfer / batchTransferTo are non-functional

CATEGORY

Correctness

LOCATION

OcicatNFT.sol, batchTransfer(...) (lines 290-296), batchTransferTo(...) (lines 298-303)

DESCRIPTION

Both call this.safeTransferFrom(...), so inside the call msg.sender is the NFT contract, not the token owner. The authorization check then fails and the call reverts unless the caller approved the contract as operator. This is functional breakage only. No token can be moved without proper authorization, so there is no theft.

IMPACT

The batch functions revert in normal use. No theft, hence Low.

PROOF OF CONCEPT

A test mints 2 NFTs and calls batchTransfer without operator approval, which reverts on the self-call authorization check.

RECOMMENDATION

Call the internal _safeTransfer(msg.sender, ...) directly rather than through this.

SMC-L-07 LOW Open

Buy/sell tax asymmetry, non-FoT router sells revert

CATEGORY

Fee and tax mechanics

LOCATION

OcicatToken.sol, _transfer(address,address,uint256) (lines 92-94)

DESCRIPTION

The tax is charged on the sender. The LP pair is whitelisted, so buys are untaxed while sells pay the 5% fee. A router that is not fee-on-transfer aware reverts on sells. The normal, fee-on-transfer-aware path works.

IMPACT

Integrations that use a non-supporting router revert on sells. The supporting path works, hence Low.

PROOF OF CONCEPT

On a fork, a buy through the pair is untaxed, a sell via the non-supporting swap function reverts on the fee-on-transfer desync, and the supporting function succeeds with the 5% tax applied.

RECOMMENDATION

Document the fee-on-transfer requirement and ensure integrators use FoT-aware router paths.

SMC AUDITS

SMC-L-08 LOW Open

Farm emergencyUnstake does not decrement totalDaoPower

CATEGORY

Accounting

LOCATION

MultiStakeRewards.sol, emergencyUnstake() (line 231)

DESCRIPTION

The function zeroes daoPower[user] but never reduces totalDaoPower (unlike withdrawAfterCooldown), so every emergency exit permanently inflates totalDaoPower. This is harmless on-chain today but corrupts any off-chain tally that reads the denominator.

IMPACT

totalDaoPower drifts upward on every emergency exit, skewing any off-chain governance tally. No on-chain effect today, hence Low.

PROOF OF CONCEPT

A test stakes, then emergencyUnstakes, and shows daoPower[user] = 0 while totalDaoPower retains the exited user's power.

RECOMMENDATION

Run totalDaoPower -= daoPower[msg.sender] before zeroing.

SMC-I-01 INFO Open

NFT owner is immutable (no rotation to multisig)

CATEGORY

Centralization

DESCRIPTION

OcicatNFT exposes no transferOwnership or renounce, so a compromised owner key cannot be rotated and the role cannot move to a multisig.

RECOMMENDATION

Add a two-step ownership transfer to a multisig.

SMC AUDITS

SMC-I-02 INFO Open

NFT owner can free-mint remaining supply and rescale power

CATEGORY

Centralization

DESCRIPTION

withdrawUnminted lets the owner mint the entire remaining supply for free, and setNFTPower rescales all holders' power. Owner-gated by design, but unbounded.

RECOMMENDATION

Bound nftPower, and reserve whitelist allocation versus public mint.

SMC-I-03 INFO Open

Admin/owner roles are single EOAs, not multisigs

CATEGORY

Centralization

DESCRIPTION

Token admin and farm/NFT owner are single externally-owned keys (one is an actively trading hot wallet), not multisigs.

RECOMMENDATION

Multisig plus timelock across the ecosystem.

SMC-I-04 INFO Open

Farm assignDaoPower and retroactive setRewardSettings

CATEGORY

Centralization / accounting

DESCRIPTION

assignDaoPower sets arbitrary governance weight, and setRewardSettings applies retroactively to already-accrued time.

RECOMMENDATION

Snapshot accrual before changing settings, and constrain assignDaoPower.

SMC AUDITS

SMC-I-05 INFO Open

NFT shared mint/claim supply, full-allocation claim

CATEGORY

Correctness / availability

DESCRIPTION

Public mint and whitelist claim share one MAX_SUPPLY with no reservation, so the public mint can starve whitelisted claimers. claim() mints the full allocation in one transaction with no partial path. The claim loop is bounded by the remaining supply (about 1,940), roughly 49M gas, within the block gas limit. Costly, but not a denial of service.

RECOMMENDATION

Reserve a claim allocation, and consider a partial or paged claim.

SMC-I-06 INFO Open

Admin setters emit no events

CATEGORY

Observability

DESCRIPTION

setWhitelist, setMintPrice, setNFTPower, and similar emit no events, reducing off-chain observability of privileged state changes.

RECOMMENDATION

Emit events on every privileged state change.

SMC-I-07 INFO Open

recoverERC20 ignores transfer return, dust fee rounding

CATEGORY

Code hygiene

DESCRIPTION

OcicatToken.recoverERC20 ignores the ERC-20 transfer return value. The 3% and 2% fees truncate to zero for dust transfers, which is economically irrelevant.

RECOMMENDATION

Check the return value, or use a safe-transfer wrapper.

Verified negative results (no finding)

Reentrancy. None exploitable. The token makes no external call on transfer. The farm guards every mutating function and updates state before transfers. The NFT receiver hook is guarded and

SMC AUDITS

ownership moves before the callback. An adversarial onERC721Received that re-enters safeTransferFrom, mint, and claim reverts on all three, with totalSupply and balances unchanged.

External interactions. No oracle, price-feed, router, flash-loan, or bridge dependency in contract logic, so there is no price-manipulation or flash-loan attack surface. The farm's only connections, the LP pair and the reward token, are fixed at construction with no setter.

Upgradeability and backdoors. No proxy or upgrade path on any contract, no hidden fund-mover, and no leaked key or secret in the public frontends.

Value conservation and theft. No stake, unstake, or restake sequence lets a user withdraw more LP than deposited. Every loss path is the user's own, never theft of another staker's LP. A victim's exit is unaffected by an attacker's state.

Access control. Every privileged function is gated to the correct role. No missing modifier, wrong role, or permissionless privileged mutation, and no uninitialized-proxy takeover.

Arithmetic. All contracts use checked arithmetic (0.8.20 or later). No unchecked blocks, no assembly, and the reward formula does not overflow across its full input domain.

LP lock. DXLock holds 76.3% of the LP until 29 April 2028. Both owner withdraw paths revert before expiry.

09

09 Remediation status

After the project addresses the findings, SMC Audits re-audits the changes and records the outcome here. No contract in scope is a proxy, so the bytecode cannot be changed in place. The recommendations should be applied in any redeployment, and the owner and holder guidance applies to the live contracts now.

ID	SEVERITY	STATUS	VERIFIED FIX
SMC-H-01	HIGH	Open	None, pending project remediation. Fix the reward decimal normalization before any further reward funding.
SMC-H-02	HIGH	Open	None, pending project remediation. Move ownership to a multisig and escrow accrued rewards.
SMC-M-01	MEDIUM	Open	None, pending project remediation. Cap the cooldown and redirect the penalty.
SMC-M-02	MEDIUM	Open	None, pending project remediation. Guard stake during cooldown and zero totalUserStaked on unstake.
SMC-M-03	MEDIUM	Open	None, pending project remediation. Pay principal unconditionally, rewards best-effort.

SMC-L-01	LOW	Open	None, pending project remediation.
SMC-L-02	LOW	Open	None (immutable). Emit Transfer for each fee leg in a future version.
SMC-L-03	LOW	Open	None, pending project remediation.
SMC-L-04	LOW	Open	None, pending project remediation. Self-inflicted only.
SMC-L-05	LOW	Open	None. Latent, no on-chain consumer of daoPower today.
SMC-L-06	LOW	Open	None (immutable). Batch functions revert, no theft.
SMC-L-07	LOW	Open	None (immutable). Supporting router path works.
SMC-L-08	LOW	Open	None, pending project remediation. Off-chain tally only.
SMC-I-01 to SMC-I-07	INFO	Open	None, pending project remediation.

10

10 Conclusion

The OCICAT token and NFT are free of the most dangerous classes: no arbitrary mint, no proxy or upgrade path, no exploitable reentrancy, no oracle or flash-loan surface, and user principal, holder balances and staked LP, cannot be stolen by an attacker. 76.3% of the LP is time-locked to 29 April 2028.

The material risk is concentrated in the staking farm. SMC-H-01 (reward over-issuance) makes the farm insolvent by design and lets the first claimant drain whatever reward reserve is funded, and SMC-H-02 lets the owner withdraw that reserve at will. The medium findings all stem from the farm's exit and cooldown accounting. Across the ecosystem, owner and admin privileges are broad and held by single externally-owned keys.

We recommend fixing the farm reward math (SMC-H-01) and exit accounting (SMC-M-01, SMC-M-02, SMC-M-03) before any further reward funding, and moving privileged roles to a multisig with a timelock. The token and NFT items are hygiene and centralization notes that do not place holder funds at risk in the live state.

An audit reduces risk. It does not guarantee the absence of every vulnerability. Users and the project should treat this report as one part of a wider security practice.

SMC AUDITS

11

11 Disclaimer

This report reflects the code reviewed at the specific version or address stated in the scope, at the time of review. It does not cover changes made after the review, external dependencies outside the stated scope, off-chain components, or economic and market conditions. This report is not financial, investment, or legal advice. SMC Audits provides no warranty, express or implied, and accepts no liability for any loss arising from the use of the audited code. A clean report is not a guarantee of safety.

SMC AUDITS · CONFIDENTIAL · SMCAUDITS.COM

SMC AUDITS