SMCAUDIT

SMART CONTRACT
SECURITY AUDIT



AUDIT REPORT

The Kingdom Coin (TKC) Security Review

PROJECT	The Kingdom Coin (TKC)
CHAIN	BNB Smart Chain (chain id 56)
CONTRACT	0x06Dc293c250e2fB2416A4276d291803fc74fb9B5
VERSION	Verified on-chain bytecode · Sourcify verified (matchId 6874264) · solc 0.8.17 · not a proxy
REPORT ID	SMC-TKC-202610-01
DATE	19 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 KingdomCoin (TKC) token contract on BNB Smart Chain at its deployed, verified address 0x06Dc293c250e2fB2416A4276d291803fc74fb9B5. TKC is a PancakeSwap V2 fee-on-transfer token: a fixed-supply BEP-20 that charges a sell fee, converts collected fees to BNB, and supports an owner-managed blacklist and wallet and fee configuration.

The review identified 12 findings: 0 critical, 2 high, 1 medium, 4 low, and 5 informational. The most significant issues are centralization controls that let the owner permanently or reversibly deny all holders the ability to sell, or asymmetrically trap buys and liquidity exits. The owner can zero the fee configuration so that every sell reverts and then renounce ownership, making the freeze permanent and irreversible (SMC-H-01); and the owner can set a 100% sell fee or blacklist the liquidity pair to freeze buys and liquidity removal while leaving sells open and draining the pool (SMC-H-02). One live condition already degrades the system: the PancakeSwap router is blacklisted on-chain today, so removeLiquidityETH reverts for every LP right now, with only the non-ETH removeLiquidity path as an escape (SMC-M-01).

Overall, the token is free of the most dangerous permissionless classes: there is no arbitrary mint (supply is fixed at construction), no proxy or upgrade path, no exploitable reentrancy on the fund path, no oracle or flash-loan surface, and no path by which an unprivileged attacker can steal holder balances or pool funds. The material risk is concentrated in broad, single-EOA owner privileges and in the design of the inline fee swap, which can be configured to revert and brick transfers.

0	2	1	4	5	12
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

As of this report, 0 of 12 findings have been resolved; the report is delivered for remediation. The contract has a live owner and is not renounced, so the centralization findings are remediable by the project through key-management and configuration changes. Any residual risk is described in the conclusion.

03

03 Project overview

TKC is a fee-on-transfer BEP-20 on BNB Smart Chain, launched in 2023, paired against WBNB on PancakeSwap V2. This audit covers the token contract only; the review confirmed no other in-scope protocol contract (staking, launchpad, treasury) is deployed on-chain under the project.

System at a glance

COMPONENT	ROLE
KingdomCoin (TKC)	Core BEP-20: fixed 240,000,000 supply, sell fee split into buy-back and operational portions.
Fee swap	On qualifying transfers the contract converts accumulated fee tokens to BNB via the PancakeSwap router and forwards to operations and buy-back wallets.
Admin / ownership	Single externally-owned owner key; no timelock, no multisig; not renounced.
Blacklist	Owner-managed address blacklist checked in _transfer.
External integrations	WBNB/TKC PancakeSwap V2 pair and router. No oracle, bridge, or upgrade path in contract logic.

Key mechanisms to note

- The sell fee is charged only on transfers to an automated-market-maker pair; buys and wallet-to-wallet transfers are untaxed. The live fee is 3% (1.5% buy-back + 1.5% operational).
- maximumTotalFee is 1000 and the fee divisor is 1000, so the owner can set the total fee up to 100%.
- The fee swap runs inline inside _transfer with no try/catch; certain owner-set configurations make it revert, which reverts the user's transfer.
- The owner can blacklist any address, including the liquidity pair, and can change the router and the fee-recipient wallets.
- Supply is fixed at construction; there is no mint, no proxy, and no pause.

SMC AUDITS

04

04 Scope

ITEM	DETAIL
Project	The Kingdom Coin (TKC)
Chain	BNB Smart Chain (chain id 56)
Language	Solidity
Contract	Deployed contract 0x06Dc293c250e2fB2416A4276d291803fc74fb9B5, verified source via Sourcify.
Commit / version	Verified live on-chain bytecode; verified source, Solidity 0.8.17, optimizer per verified metadata. Not a proxy.

Addresses

COMPONENT	ADDRESS
KingdomCoin (TKC) token	0x06Dc293c250e2fB2416A4276d291803fc74fb9B5
WBNB/TKC LP pair (PancakeSwap V2)	0xBeE567474f87F7725791F2872D165FB69e0bBcDd
PancakeSwap V2 router	0x10ED43C718714eb63d5aA57B78B54704E256024E
Owner	0xA6D70ed7c08A825403156d58bc67472F9c717349
operationsWallet	0x85353f651D8e09F4490f36E911Cd2c5c6b016385
buyBackWallet	0xE428fAC3ED6595749afe46e73Af7de3Ed627f2D2

Files reviewed

FILE	LINES	NOTES
KingdomCoin.sol	~996	Single-file BEP-20 + fee-on-transfer + fee swap; includes Context, Ownable, ERC20 base, SafeMath libraries, Uniswap V2 interfaces, and the KingdomCoin contract.

Out of scope

The PancakeSwap V2 pair and router internals (third-party, reviewed only for how TKC interacts with them), the off-chain frontends and project website, deployment and key management, and market or economic conditions. The advertised wallet, staking and launchpad features have no in-scope deployed contract and are not part of this review.

SMC AUDITS

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 contract does, 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. Proofs were run against the live bytecode on a BNB-chain fork; no live mainnet transactions were made.
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

SMC AUDITS

07

07 Findings summary

ID	TITLE	SEVERITY	STATUS
SMC-H-01	Owner can permanently freeze all sells (honeypot), irreversibly after renounce	HIGH	Open
SMC-H-02	Owner can set a 100% sell fee, or blacklist the pair to freeze buys/exits and drain the pool	HIGH	Open
SMC-M-01	Router is blacklisted on-chain: removeLiquidityETH reverts for all LPs today	MEDIUM	Open

SMC-L-01	Blacklist checks only the sender; recipients still receive and a buyBackWallet bypass exists	LOW	Open
SMC-L-02	maxSellTransactionAmount is never enforced and cannot be set to a usable value	LOW	Open
SMC-L-03	Native BNB sent to the contract is permanently stuck (no rescue)	LOW	Open
SMC-L-04	Dead launch-gate flags are set but never read	LOW	Open
SMC-I-01	updateWallets never revokes the old wallets' fee exemption	INFO	Open
SMC-I-02	Event-emission bugs misreport fee and wallet changes	INFO	Open
SMC-I-03	Zero-amount transferFrom enables address-poisoning spoofed Transfer events	INFO	Open
SMC-I-04	Privileged roles are a single EOA, not a multisig/timelock	INFO	Open
SMC-I-05	Redundant SafeMath on Solidity 0.8.x	INFO	Open

0	2	1	4	5	12
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

SMC AUDITS

08

08 Detailed findings

PoC note: fork tests select a BNB-chain fork and run against the live bytecode at the deployed address. No live mainnet transaction was made. The full suite is 15/15 passing. The full tests are provided in the engagement materials.

SMC-H-01 HIGH Open

Owner can permanently freeze all sells (honeypot), irreversibly after renounce

CATEGORY

Centralization / denial of service

LOCATION

KingdomCoin.sol: _transfer(...) fee-swap call (line ~936), swapAndSendDividends(uint256) (lines 970-992), updateFees(uint256,uint256) (line 869), updateSwapTokensAtAmount(uint256) (line 800)

DESCRIPTION

The fee swap runs inline inside _transfer with no try/catch, so any revert inside it reverts the user's transfer. Several owner-reachable configurations make it revert: updateFees(0,0) makes swapAndSendDividends divide by totalFees == 0; operationalFee == 0 sends 0 tokens to the router, which reverts with INSUFFICIENT_INPUT_AMOUNT; updateSwapTokensAtAmount(0) triggers the swap on an empty balance. The swap fires on a sell once the contract holds at least swapTokensAtAmount fee tokens, so after one of these configurations is set, every sell and non-exempt wallet transfer reverts while buys continue to work. The owner can then call renounceOwnership(), after which no one can restore the configuration and the freeze is permanent. The reverting swap never drains the contract balance, so the trap is self-sustaining.

IMPACT

All holders are permanently unable to sell or move their tokens, while buys keep working, a one-way honeypot. Realized worst case is a total, irreversible freeze of every holder's position. The trigger is the owner only (no unprivileged attacker can cause it), but it requires no cooperation from holders and cannot be undone once ownership is renounced.

PROOF OF CONCEPT

On a BSC fork at the live bytecode, the owner zeroes the fee configuration, renounces ownership, and every subsequent sell reverts while buys still succeed.

```
function test_ESC1_permanent_honeypot_after_renounce() public {
    uint256 got = _buy(5 ether);
    // arm: push the contract over its swap threshold, then zero the fees
    uint256 threshold = TKC.swapTokensAtAmount();
    if (TKC.balanceOf(address(TKC)) < threshold) {
        vm.prank(user);
        TKC.transfer(address(TKC), threshold + 1e21);
    }
    vm.prank(OWNER); TKC.updateFees(0, 0);
    vm.prank(OWNER); TKC.renounceOwnership();
    assertEq(TKC.owner(), address(0), "ownership renounced - no one can undo it");
    uint256 got2 = _buy(1 ether);
    assertGt(got2, 0, "buys still work");
    _approveRouter(got / 4);
    vm.expectRevert();
    _swapSell(got / 4);          // every sell reverts, forever
    assertGe(TKC.balanceOf(address(TKC)), threshold, "trap is self-sustaining");
}
```

The operationalFee == 0 and swapTokensAtAmount == 0 variants are proven by test_K9_operationalFeeZero_bricks_sells and test_K9_updateFeesZero_bricks_all_sells.

RECOMMENDATION

Wrap the router swap in try/catch and skip the swap when the amount or total fee is zero, so a fee-swap failure can never block a user transfer. Validate fee setters (reject a zero operational or total fee that would brick the swap). Place ownership behind a multisig and timelock, and if ownership is renounced, require it to be renounced only from a known-safe configuration.

SMC AUDITS

SMC-H-02 HIGH Open

Owner can set a 100% sell fee, or blacklist the pair to freeze buys/exits and drain the pool

CATEGORY

Access control / centralization

LOCATION

KingdomCoin.sol: updateFees(uint256,uint256) (line 869), _transfer(...) fee branch (lines 958-964), blackList(address[],bool) (line 820), _transfer(...) blacklist branch (lines 911-915)

DESCRIPTION

Two owner powers let the owner trap or take holder and LP value. **100% fee.** maximumTotalFee is 1000 and the fee is amount * totalFees / 1000, so updateFees can take up to 100% of a sell. At 100% the sell reverts (the router receives 0); at 99.9% the seller receives almost nothing. **Pair blacklist.** blackList can flag the PancakeSwap pair. Because _transfer blocks transfers whose from is blacklisted, this reverts every buy (pair to buyer) and every liquidity removal (pair to LP), while sells (user to pair) still succeed. Holders can then only dump into a pool the owner drains, and the owner can still exit via the buyBackWallet blacklist exemption (see SMC-L-01).

IMPACT

The owner can, at will, make sells return nothing (fee) or freeze buys and liquidity exits while keeping the sell side open and draining pool BNB (pair blacklist). Both reach all holders and LPs. Owner-gated; no unprivileged attacker can trigger either.

PROOF OF CONCEPT

```
function test_K1_hundred_percent_fee_blocks_sell() public {
    uint256 got = _buy(3 ether);
    vm.prank(OWNER); TKC.updateFees(500, 500);   // totalFees = 1000 = 100%
    assertEq(TKC.totalFees(), 1000, "fee set to 100%");
    _approveRouter(got / 2);
    vm.expectRevert();
    _swapSell(got / 2);                          // sell reverts
}

function test_ESC2_pair_blacklist_asymmetric_rug() public {
    uint256 tkc = _buy(10 ether);
    vm.startPrank(user);
    TKC.approve(address(ROUTER), tkc);
    (,, uint256 liquidity) = ROUTER.addLiquidityETH{value: 5 ether}(address(TKC),
tkc, 0, 0, user, block.timestamp);
    vm.stopPrank();
    address[] memory list = new address[](1); list[0] = PAIR;
    vm.prank(OWNER); TKC.blackList(list, true);
    // buys revert, LP removal reverts, but sells still work
    vm.prank(user); vm.expectRevert();
    ROUTER.swapExactETHForTokensSupportingFeeOnTransferTokens{value: 1 ether}(0,
_path(WBNB, address(TKC)), user, block.timestamp);
    vm.prank(user); IERC20(PAIR).approve(address(ROUTER), liquidity);
    vm.prank(user); vm.expectRevert();
    ROUTER.removeLiquidity(address(TKC), WBNB, liquidity, 0, 0, user,
block.timestamp);
    assertGt(_sell(TKC.balanceOf(user) / 4), 0, "sells still work while buys and LP
```

```
exits are frozen");
}
```

A 90% fee returning a fraction of the honest proceeds is quantified by test_K1_near_total_fee_drains_seller (0.976 BNB at 3% vs 0.101 BNB at 90% on a half-sell).

RECOMMENDATION

Cap maximumTotalFee at a safe value (for example, ≤ 100 = 10%). Do not allow the AMM pair to be blacklisted or fee-exempt-bypassed; block both transfer directions in the blacklist check and remove the buyBackWallet carve-out. Place privileged roles behind a multisig and timelock.

SMC AUDITS

SMC-M-01 MEDIUM Open

Router is blacklisted on-chain: removeLiquidityETH reverts for all LPs today

CATEGORY

Denial of service (live condition)

LOCATION

KingdomCoin.sol: _transfer(...) blacklist branch (lines 911-915); live blacklist state for the PancakeSwap router 0x10ED43C7...

DESCRIPTION

The PancakeSwap V2 router is currently blacklisted in the token (set on 2023-05-14). removeLiquidityETH and removeLiquidityETHSupportingFeeOnTransferTokens perform a final token transfer whose from is the router, which the blacklist check reverts. As a result, these standard liquidity-removal paths revert for every LP right now. The router wraps that transfer in TransferHelper, so the surfaced error is TransferHelper: TRANSFER_FAILED, masking the underlying Account Blacklisted revert.

IMPACT

LPs cannot remove liquidity through the common removeLiquidityETH path. The impact is bounded to a Medium because the non-ETH removeLiquidity(TKC, WBNB, ...) path still works (the pair sends tokens directly to the LP), giving LPs an escape. If the owner also blacklists the pair, that escape closes too (see SMC-H-02).

PROOF OF CONCEPT

```
function test_K10_router_blacklisted_breaks_removeLiquidityETH_leg() public {
    assertTrue(TKC.isBlacklisted(address(ROUTER)), "router is blacklisted live");
    vm.prank(PAIR); TKC.transfer(address(ROUTER), 1000e18);   // pair -> router
allowed
    vm.prank(address(ROUTER));
    vm.expectRevert(bytes("Account Blacklisted"));
    TKC.transfer(user, 1000e18);                              // router -> user leg
reverts
}
```

test_K10_e2e_removeLiquidityETH_reverts_removeLiquidity_works confirms end to end that removeLiquidityETH reverts (TransferHelper masks the blacklist revert) while the non-ETH removeLiquidity succeeds and returns TKC + WBNB.

RECOMMENDATION

Do not blacklist AMM infrastructure (router or pair). Narrow the blacklist so it cannot break routed transfers for all users, and surface a clear revert reason.

SMC-L-01 LOW Open

Blacklist checks only the sender; recipients still receive and a buyBackWallet bypass exists

CATEGORY

Access control / correctness

LOCATION

KingdomCoin.sol: _transfer(...) blacklist branch (lines 911-915)

DESCRIPTION

The revert fires only when from is blacklisted and to != buyBackWallet. A blacklisted recipient still receives tokens (the inner require checks from), and a blacklisted sender can still move tokens to buyBackWallet. The blacklist therefore only blocks transfers from a blacklisted account to a non-buyBackWallet address.

```
if (_isBlacklist[from] || _isBlacklist[to]) {
    if (to != buyBackWallet) {
        require(!_isBlacklist[from], "Account Blacklisted");
    }
}
```

IMPACT

The blacklist is an owner tool, and its asymmetry weakens the owner's own control (a frozen account can still receive and can still exit to buyBackWallet). No third-party fund loss.

PROOF OF CONCEPT

test_K4_blacklisted_recipient_still_receives (a blacklisted recipient receives) and test_K4_blacklisted_sender_bypass_via_buyBackWallet (a blacklisted sender is blocked to a normal address but moves tokens to buyBackWallet).

RECOMMENDATION

Block both directions explicitly, e.g. require(!_isBlacklist[from] && !_isBlacklist[to], "Blacklisted"), and reconsider the buyBackWallet exemption.

SMC AUDITS

SMC-L-02 LOW Open

maxSellTransactionAmount is never enforced and cannot be set to a usable value

CATEGORY

Correctness / dead control

LOCATION

KingdomCoin.sol: maxSellTransactionAmount (line 735), updateMaximumSales(uint256) (line 806); _transfer never reads the variable

DESCRIPTION

maxSellTransactionAmount is declared and settable but is never read in _transfer, so the advertised anti-whale sell cap does nothing. In addition, updateMaximumSales requires the new value to exceed 20% of total supply while its revert message says 5%, and the shipped value (10,000,000) is below that 20% floor, so the launch value can never be restored. The setter is both inconsistent and, combined with the unused variable, non-functional.

IMPACT

A protection users may rely on does not exist; no fund loss.

PROOF OF CONCEPT

test_K2_K6_sell_limit_unusable: the shipped limit is below the setter's 20% floor; updateMaximumSales of a sane value reverts; and an oversized sell is not blocked by the limit.

RECOMMENDATION

Either enforce the limit in _transfer on sells, or remove the variable and its setter to avoid misleading holders. Fix the inconsistent bound and message.

SMC AUDITS

SMC-L-03 LOW Open

Native BNB sent to the contract is permanently stuck

CATEGORY

Fund lock

LOCATION

KingdomCoin.sol: receive() external payable (line 797); no withdraw/rescue function exists

DESCRIPTION

The contract has a payable receive() and accepts BNB, but exposes no function that transfers BNB (or arbitrary tokens) out. Any native BNB sent to the contract is permanently unrecoverable.

IMPACT

Stuck funds, no theft. The contract currently holds a non-trivial BNB and token balance that cannot be recovered.

PROOF OF CONCEPT

test_K11_bnb_sent_to_contract_is_stuck: BNB sent to the contract is accepted and held, with no path to withdraw it.

RECOMMENDATION

Add an owner-restricted rescue function for native BNB and non-fee tokens, or remove receive() if the contract is not meant to hold BNB.

SMC-L-04 LOW Open

Dead launch-gate flags are set but never read

CATEGORY

Dead code / correctness

LOCATION

KingdomCoin.sol: tradingEnabled (line 745), canTransferBeforeTradingIsEnabled (line 746); set in the constructor (lines 791-792), never read

DESCRIPTION

tradingEnabled and the canTransferBeforeTradingIsEnabled mapping are set at construction but never read anywhere in the contract. There is no actual pre-trading gate; the names imply a launch control that does not exist.

IMPACT

No runtime effect (trading is live); indicates removed or incomplete logic and can mislead reviewers and holders.

RECOMMENDATION

Remove the unused state, or implement a real launch gate if one is intended.

SMC AUDITS

SMC-I-01 INFO Open

updateWallets never revokes the old wallets' fee exemption

CATEGORY

Fee, tax, and reflection mechanics

DESCRIPTION

updateWallets excludes the new operations and buy-back wallets from fees but never revokes the exemption of the wallets it replaces, so a retired or compromised wallet keeps selling fee-free.

RECOMMENDATION

Un-exclude the old wallets on rotation.

SMC-I-02 INFO Open

Event-emission bugs misreport fee and wallet changes

CATEGORY

Best practice / code quality

DESCRIPTION

FeesUpdated is emitted with its two fee arguments swapped; BuyBackWalletUpdated logs the new operations wallet instead of the new buy-back wallet; UpdateMaximumSell is emitted before the assignment, so it logs the old value. Off-chain indexers will misreport these changes. Proven by test_K13_feesUpdated_event_args_swapped.

RECOMMENDATION

Fix the event arguments and emit after the state change.

SMC-I-03 INFO Open

Zero-amount transferFrom enables address-poisoning spoofed Transfer events

CATEGORY

Token and supply logic

DESCRIPTION

A zero-amount transferFrom(victim, attacker, 0) needs no allowance (the base path's allowance.sub(0) does not underflow) and the zero-amount branch emits a Transfer(victim, attacker, 0) event, enabling address-poisoning spoofing. No value moves. Proven by test_K14_zero_amount_transferFrom_poisoning.

RECOMMENDATION

This is an ecosystem-wide ERC-20 nuance; optionally reject zero-value transfers or document it.

SMC AUDITS

SMC-I-04 INFO Open

Privileged roles are a single EOA, not a multisig/timelock

CATEGORY

Privilege and centralization risk

DESCRIPTION

The owner is a single externally-owned key, not renounced and with no timelock, holding the fee, blacklist, router, and wallet powers behind the High findings.

RECOMMENDATION

Move ownership to a multisig with a timelock.

SMC-I-05 INFO Open

Redundant SafeMath on Solidity 0.8.x

CATEGORY

Best practice / code quality

DESCRIPTION

The SafeMath / SafeMathInt / SafeMathUint libraries are redundant under Solidity 0.8.x checked arithmetic and cost gas.

RECOMMENDATION

Remove them and use native arithmetic.

SMC AUDITS

Verified negative results (no finding)

- Supply / mint: supply is fixed at 240,000,000 at construction; there is no mint function and no owner path to inflate supply.
- Reentrancy: the base _transfer follows checks-effects-interactions; the fee swap is guarded by a swapping flag; no exploitable reentrancy on the fund path.
- Access control: every state-changing admin function is gated by onlyOwner; no missing modifier or permissionless privileged mutation was found. There is no proxy or initializer takeover surface.
- Arithmetic: Solidity 0.8.17 checked arithmetic; no unchecked fund-path math; no overflow.
- External interactions: no oracle, price feed, or bridge in contract logic, no price-manipulation or flash-loan surface.
- MEV / fee-swap sandwich: the inline fee swap uses amountOutMin = 0 on the full balance, but a sandwich of that swap is net-negative to an attacker at the live pool depth (~7.55M TKC / ~90 WBNB) and the 3% fee; it is not a profitable exploit. Retained only as a hardening note.
- Permissionless theft: no unprivileged attacker can steal holder balances or pool funds; every fund-impacting path requires the owner.

09

09 Remediation status

After the project addresses the findings, SMC Audits re-audits the changes and records the outcome here. The contract has a live owner and is not renounced, so the centralization findings are remediable through configuration and key-management changes, and the fee-swap and blacklist logic would require a redeployment to fix in code.

ID	SEVERITY	STATUS	VERIFIED FIX
SMC-H-01	HIGH	Open	Pending project remediation
SMC-H-02	HIGH	Open	Pending project remediation
SMC-M-01	MEDIUM	Open	Pending project remediation (owner can un-blacklist the router today)
SMC-L-01	LOW	Open	Pending project remediation
SMC-L-02	LOW	Open	Pending project remediation
SMC-L-03	LOW	Open	Pending project remediation
SMC-L-04	LOW	Open	Pending project remediation
SMC-I-01..I-05	INFO	Open	Pending project remediation

SMC AUDITS

10

10 Conclusion

The TKC token is free of the most dangerous permissionless classes: no arbitrary mint (supply is fixed), no proxy or upgrade path, no exploitable reentrancy, no oracle or flash-loan surface, and no path by which an unprivileged attacker can steal holder balances or pool funds.

The material risk is centralization. The owner can permanently freeze all sells and renounce ownership to make the freeze irreversible (SMC-H-01), and can set a 100% fee or blacklist the pair to freeze buys and liquidity exits while draining the pool (SMC-H-02). One live condition already degrades the system: the router is blacklisted on-chain, so removeLiquidityETH reverts for every LP today, with only the non-ETH removeLiquidity path as an escape (SMC-M-01). The remaining findings are owner-tool weaknesses, dead controls, stuck funds, and reporting bugs. We recommend fixing the inline fee swap

(wrap in try/catch, skip on zero amount or fee), capping the maximum fee, not blacklisting AMM infrastructure, correcting the blacklist asymmetry, and moving privileged roles to a multisig with a timelock before any further reliance on the token.

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.

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