SMC AUDIT

SMART CONTRACT
SECURITY AUDIT



AUDIT REPORT

TWD (THE WORD) Token Security Review

PROJECT	TWD / THE WORD ("THE WORD")
CHAIN	BNB Chain (BSC, chain id 56)
CONTRACT	0xf00cD9366A13e725AB6764EE6FC8Bd21dA22786e
VERSION	Verified source (Sourcify exact match, matchId 30454950) · Solidity 0.8.15
REPORT ID	SMC-TWD-202609-01
DATE	2 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 TWD / THE WORD smart contract on BNB Chain at address 0xf00cD9366A13e725AB6764EE6FC8Bd21dA22786e, using the source verified on Sourcify (an exact match of both creation and runtime bytecode).

The review ran on 30 September 2026 and covered the deployed CoinManufactory contract in full, including its reflection, fee, swap-back, allowance and administrative logic, and cross-checked it against live on-chain state and the complete administrative history. The manual review, in four passes (line-by-line, attacker view, owner powers and centralization, severity consolidation) plus a diff against sibling CoinManufactory deployments, was supported by cast (Foundry 1.8.3) for live state, storage slots and archive history; Foundry fork tests at block 124,945,000 for every finding, plus 3 fuzz tests (256 runs each) and 5 invariants (24 runs x 25 calls); Slither 0.11.6 and Aderyn 0.6.8, with every medium-or-higher result triaged; and Sourcify for source and metadata verification. All 28 tests pass.

The review identified 11 findings: 0 critical, 1 high, 1 medium, 1 low, and 8 informational. No issue lets an outside attacker take user funds. The most significant observation is that the renounced ownership does not make the token admin-free: the developer fee receiver, a live EOA that can no longer be replaced, can stop every sell on the TWD/WBNB pair by giving itself contract code through EIP-7702, and can lift the stop whenever it wants (SMC-H-01, High). The deployer can also withdraw 30% of supply from an expired PinkLock and sell it fee-free, which in our fork test drained 66.6% of the pool's BNB (SMC-M-01, Medium). The remaining code-level findings are an approve race (SMC-L-01, Low) and eight informational items.

Overall, the token behaves as written under its frozen configuration: fuzz and invariant tests found no honeypot, fee drift or accounting error in normal trading. Its integrity is contingent on the developer wallet 0xd769...b003 never holding code that rejects a 2300-gas BNB transfer, a condition that is true today, that no one but that wallet's key holder can guarantee, and that can be reversed in one transaction.

0	1	1	1	8	11
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

As of this report, 0 of 11 findings have been resolved and verified by SMC Audits. The contract is immutable and ownership is renounced, so the recommendations are advisory: they apply to the project team, token holders, integrators and any future redeployment rather than to the live bytecode.

03

03 Project overview

TWD (THE WORD, on-chain name "THE WORD") is a reflection and liquidity-generating BEP-20 token on BNB Chain. This audit covers the single deployed contract (CoinManufactory) that implements balances through reflection accounting, a 7% transfer tax (1% reflection to holders, 1% auto-liquidity, 5% developer fee paid in BNB), an automatic swap-back through PancakeSwap V2, and owner-only fee and limit controls. The contract is an unmodified CoinManufactory token-generator template compiled with solc 0.8.15: its source is byte-identical to another CoinManufactory deployment (0x785dB0eb0361f3Da846613a74E0b243bE9A0Da99), so the project-specific surface is limited to constructor parameters and live state. It is not a proxy and is not upgradeable.

System at a glance

COMPONENT	ROLE
CoinManufactory token contract	Reflection-based balances (_rOwned / _rTotal), transfer / transferFrom, allowances, 100,000,000,000 TWD supply (18 decimals), no mint.
Fee and swap-back logic (in-contract)	Every transfer where neither side is fee-exempt pays 7%: 1% reflected to holders, 6% kept by the contract. When a transfer goes to the pair and the contract holds at least 20,000,000 TWD, it sells 18,333,333 TWD for BNB, pays 50/55 of the BNB to developerAddress and pairs the rest with 1,666,667 TWD as liquidity. Burn fee is 0.
Admin / ownership	owner() held every setter (fees, fee receivers, threshold, max tx, swap toggle, fee and reward exclusions, presale). Ownership was renounced to address(0) on 3 February 2023; every owner function is dead. The developer fee receiver is the deployer EOA 0xd76914B765625475D2524EbbC8503457CCEBb003 and can no longer be changed.
External integrations	PancakeSwap V2 router 0x10ED...024E and pair TWD/WBNB 0x4F61...1e83 (created in the constructor). LP locker (Mudra), token locker (PinkLock02), DEX contracts and market conditions are out of scope, but their live state was checked.

Key mechanisms to note

- Renouncing ownership did not remove every live privilege. owner() returns address(0) (renounce 3 February 2023), and the only owner calls ever made were setDeveloperFee(5) three minutes earlier and the renounce itself. Two privileges survive because they never depended on owner(): the developer fee receiver, paid with a 2300-gas transfer inside every sell that triggers the swap-back, and the deployer's token lock on PinkLock02. These are the basis of SMC-H-01 and SMC-M-01.
- The 7% tax applies to buys, sells and wallet-to-wallet transfers. A 1,000,000 TWD wallet transfer delivers 930,000 TWD, books 10,000 TWD as reflection and adds 60,000 TWD to the contract. The contract, 0x...dEaD and the deployer are fee-exempt. There is no max-wallet, cooldown, blacklist, whitelist or trading switch, and _maxTxAmount equals total supply.
- Liquidity is locked; supply is not. 85.46% of the LP tokens are locked in the Mudra locker until 30 January 2100 and 14.53% sit at address(0). In contrast, PinkLock02 lock 1069189 holds

30,000,000,000 TWD (30% of supply) for the deployer, and its unlock date of 30 January 2025 has passed. Token trackers still show these tokens as locked.

04

04 Scope

ITEM	DETAIL
Project	TWD / THE WORD ("THE WORD")
Chain	BNB Chain (BSC, chain id 56)
Language	Solidity
Contract	0xf00cD9366A13e725AB6764EE6FC8Bd21dA22786e (verified source via Sourcify)
Commit / version	Deployed bytecode; Sourcify exact match (matchId 30454950, verified 2026-06-02). Deployed at block 25225199 on 2023-01-30 by 0xd769…b003. Not a proxy.
Compiler	solc 0.8.15+commit.e14f2714; optimizer disabled (runs 200); EVM version london

Files reviewed

FILE	LINES	NOTES
CoinManufactory.sol	1874	Entire contract: IERC20, IERC20Metadata, Address, Context, Ownable, ERC20, Uniswap V2 factory/pair/router interfaces, CoinManufactory. SHA-256 d73146a7…3223b.

Out of scope

Off-chain components, the project website and front end, and the separately deployed TWDstaking (0xb178…7575) and SMC (0xe8d8…5be8) contracts that hold TWD. Liquidity pools, DEX pairs, routers, token and LP lockers, and market or economic conditions. The PinkLock02 and Mudra lock states were read on-chain because they change the risk to holders; PinkLock02 is verified and its unlock logic was reviewed; the Mudra locker is not verified, so its migrator path could not be reviewed. Deployment process and key management history. No evidence gap affects a Critical, High, or Medium conclusion; the complete owner history was rebuilt from archive state because the setters emit no events.

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.
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.

SMC AUDITS

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-H-01	Immutable developer fee receiver can halt every sell on the main pair	HIGH	Open
SMC-M-01	Expired PinkLock releases 30% of supply to the deployer at will	MEDIUM	Open
SMC-L-01	ERC-20 approve race condition (SWC-114)	LOW	Open
SMC-I-01	Privileged functions rely entirely on renounced ownership	INFO	Acknowledged
SMC-I-02	Fee save and restore ignores the burn fee and restores stale fees	INFO	Open
SMC-I-03	Swap-back reverts for valid fee configurations	INFO	Open
SMC-I-04	Fee-on-transfer on every transfer breaks standard integrations	INFO	Open
SMC-I-05	Reflection side effects: skimmable pair balance, stranded reflections, rounding	INFO	Open
SMC-I-06	BNB accumulates in the token contract with no way out	INFO	Open
SMC-I-07	Code quality and ERC-20 edge cases	INFO	Open
SMC-I-08	Swap-back uses zero slippage limits	INFO	Open

0	1	1	1	8	11
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

SMC AUDITS

08

08 Detailed findings

SMC-H-01 HIGH Open

Immutable developer fee receiver can halt every sell on the main pair

CATEGORY

Privilege and centralization risk; denial of service

LOCATION

_transfer(address,address,uint256) L1566-1575, swapTokens(uint256) L1743-1757, transferToAddressETH(address,uint256) L1869-1873, setDeveloperAddress(address) L1419-1421

DESCRIPTION

When a transfer goes to uniswapV2Pair and the contract holds at least minimumTokensBeforeSwap (20,000,000 TWD), _transfer runs swapTokens before moving the seller's tokens. swapTokens sells the inventory and pays developerAddress with `recipient.transfer(amount)`, which forwards only 2300 gas and reverts the whole sell if the receiver rejects the payment. developerAddress is the deployer EOA 0xd769...b003 and can no longer be changed, and the swap-back cannot be turned off, because both setters are onlyOwner and ownership is renounced. Since BSC's Pascal hard fork (20 March 2025, BEP-441), an EOA can give itself contract code with one EIP-7702 authorization and remove it again later. If the developer wallet delegates to code whose receive reverts or needs more than 2300 gas, every sell into the pair reverts. Buys and wallet transfers keep working and keep adding tax to the inventory, so the stop lasts until the key holder removes the delegation. This can happen on purpose, or by accident through a phished key delegated to a "sweeper", a pattern already live on BSC, or through a smart-account delegate whose receive writes storage.

IMPACT

Holders of the only liquid pool cannot sell while buyers can still buy, which is the honeypot pattern. The second pool (TWD/USDT 0x7f89...0211) holds 0.0079 USDT, so there is no practical exit. The switch is invisible to scanners that treat owner() == 0 as "no admin risk", and only the key holder can reverse it. Combined with SMC-M-01, the developer can clear the delegation, unlock and dump 30,000,000,000 TWD, and re-arm it in one bundle, leaving other holders unable to sell. Rated High because the impact is serious and it needs only one transaction from a wallet the project controls today, consistent with how SMC rates comparable owner-controlled sell blocks. It is not Critical: it is not armed today (cast code 0xd769...b003 returns 0x), and it moves no holder's tokens.

PROOF OF CONCEPT

On a fork of BSC at block 124,945,000, vm.etch(dev, 0xef0100 ‖ delegate) writes exactly the code an EOA has after a 7702 authorization. All 7 tests in H01_DevReceiverSellHalt.t.sol pass. With no delegation, alice sells 282,819,872 TWD for 0.862 BNB and the swap-back pays the developer. With a reverting delegate, a full sell, a 1 TWD sell and a sell from a fee-exempt seller all revert while buys and transfers still work; the contract inventory grows from 12,907,811 to 43,200,709 TWD during the halt, and after vm.etch(dev, "") the same sell succeeds again. A sweeper delegate and a wallet delegate that writes storage in receive both halt sells; an empty receive() passes. The live BSC sweeper delegate 0xa191...79df halts sells. Selling through the USDT pair returns 0.0079 USDT. After 10 BNB of buys during the halt, the developer exits with

42.03 BNB, leaving 17.15 BNB, and holders still cannot sell. A live mainnet eth_call confirms BSC runs 7702 code. A full test and the live probe output are provided in the engagement materials.

RECOMMENDATION

Live contract: immutable; no code fix is possible. The developer wallet owner should publicly commit never to attach 7702 code to 0xd769...b003 and keep that key in cold storage; holders and trackers should treat the wallet as an admin key and monitor eth_getCode on it. Future versions: pay fee receivers with a pull pattern, or with call whose failure is tolerated, never with transfer / send inside the user transfer path; keep the swap-back switchable by a timelocked multisig rather than making it permanent by renouncing.

SMC AUDITS

SMC-M-01 MEDIUM Open

Expired PinkLock releases 30% of supply to the deployer at will

CATEGORY

Privilege and centralization risk; token and supply logic

LOCATION

Live state: PinkLock02 0x4079...1BBE lock 1069189; _isExcludedFromFee[owner()] set in constructor L1329

DESCRIPTION

On 30 January 2023 the deployer locked 30,000,000,000 TWD (30% of supply) in PinkLock02 (lock 1069189, description "Body of Christ lock"). The lock is a normal lock (tgeBps = 0) whose unlock date tgeDate = 1738229760 (30 January 2025) has passed, and unlockedAmount = 0. PinkLock02's unlock only checks owner == msg.sender, block.timestamp >= tgeDate and unlockedAmount == 0, so the deployer can withdraw the full amount with one transaction. The deployer is still fee-exempt (the constructor exempts the initial owner, and renouncing does not remove it), so the tokens arrive and can be sold without the 7% tax.

IMPACT

A full dump into the pool (48.18 WBNB / 14.99B TWD) returns 32.10 BNB, 66.6% of the pool's BNB, and cuts the value of every other holding by about 88%. This is not a defect in the TWD code, and the tokens belong to the deployer, which is why it is not rated High. It is rated Medium because it needs no special condition, it is large, and holders are shown these tokens as "locked" by third-party trackers even though they are not.

PROOF OF CONCEPT

getLocksForToken on PinkLock02 returns lock 1069189 (owner 0xd769...b003, 30,000,000,000e18, unlock 1738229760, "Body of Christ lock"). Both tests in M01_PinkLockUnlockDump.t.sol pass. The lock has been unlockable for 608 days. The deployer receives 30,000,000,000 TWD from PinkLock with no tax; with the pool at 48.18 WBNB, the dump returns 32.10 BNB (6,662 bps of the pool), and the quote for 1% of supply falls by 8,839 bps. A full test is provided in the engagement materials.

RECOMMENDATION

Live contract: no code change applies. The deployer should relock the tokens with a new, published unlock or vesting schedule, or send them to 0x...dEaD. Holders and listing sites should treat lock 1069189 as unlocked supply held by the deployer. Future versions: vest insider allocations linearly, and remove the fee exemption from the deployer when ownership is renounced.

SMC-L-01 LOW Open

ERC-20 approve race condition (SWC-114)

CATEGORY

Approvals and allowance handling

LOCATION

approve(address,uint256) L604-612, transferFrom(address,address,uint256) L627-644 (mitigations increaseAllowance L658-669, decreaseAllowance L685-700)

DESCRIPTION

approve overwrites the allowance. If an owner changes a non-zero allowance to another non-zero value, the spender can front-run the change, spend the old allowance and then spend the new one.

IMPACT

A spender can take up to the sum of both allowances. It needs a specific victim action plus a transaction-ordering advantage, and the contract already offers increaseAllowance / decreaseAllowance, hence Low.

PROOF OF CONCEPT

test_approveRace: Alice (150 TWD) approves Bob for 100, then tries to lower it to 50. Bob front-runs transferFrom(alice, bob, 100) and then spends the new 50. Result: Bob pulled 150 gross (139.5 net after the 7% tax); Alice is left with reflection dust instead of 100.

RECOMMENDATION

Immutable, no code fix. Holders and integrators should use increaseAllowance / decreaseAllowance, or set the allowance to 0 before a new value. Future versions: keep these helpers, or support EIP-2612 permits with exact amounts.

SMC AUDITS

SMC-I-01 INFO Acknowledged

Privileged functions rely entirely on renounced ownership

CATEGORY

Privilege and centralization risk

LOCATION

fee setters L1375-1413, setReflectionFee L1463-1465, address setters L1415-1425, setNumTokensSellToAddToLiquidity L1427-1432, setSwapAndLiquifyEnabled L1434-1437, setMaxTxAmount L1439-1441, exemptions L1447-1453, reward exclusion L1518-1538, presale L1856-1867, transferOwnership L446-452

DESCRIPTION

Every administrative function is onlyOwner, has no bounds and emits no event. With a live owner they would allow fee seizure, a full transfer freeze and a sell halt. owner() is address(0), and _owner can only be written through onlyOwner paths, so these functions can never run again.

IMPACT

None today (status Acknowledged: mitigated by the renounce). This is the main safety fact of the contract and should be re-verified independently by any reader (cast call owner()). It does not cover the non-owner privileges in SMC-H-01 and SMC-M-01. If an owner were active, the fee setters could push total fee above 100 (every taxed transfer reverts) or seize up to 100%, setMaxTxAmount(0) would freeze all non-owner transfers, and threshold/address setters would halt the swap-back; each was reproduced with a simulated owner on the fork.

PROOF OF CONCEPT

test_allOwnerFunctionsRevert_liveState passes: all 18 owner entry points revert with "Ownable: caller is not the owner". owner() returns address(0). Admin history was rebuilt in full: OwnershipTransferred 0x0 to deployer (block 25225199) and deployer to 0x0 (block 25345588), and all 75 deployer transactions before the renounce, which include exactly one other owner call, setDeveloperFee(5).

RECOMMENDATION

No action on the live contract. Future versions: cap fees in basis points in the setter, bound limits and thresholds, emit events for every configuration change, and use a timelocked multisig instead of renouncing if parameters still need to be managed.

SMC AUDITS

SMC-I-02 INFO Open

Fee save and restore ignores the burn fee and restores stale fees

CATEGORY

Fee, tax, and reflection mechanics

LOCATION

removeAllFee() L1826-1844, restoreAllFee() L1846-1854, _tokenTransfer L1586-1601, presale(bool) L1856-1867

DESCRIPTION

Fee-exempt transfers zero the fees in storage and restore them afterwards. removeAllFee returns early when _combinedLiquidityFee == 0 && _reflectionFee == 0, without checking _burnFee and without saving the current values. An exempt transfer then still pays the burn fee, and restoreAllFee writes the old _previous* values over the current configuration, which the setters never update.

IMPACT

Not reachable today: the configuration is frozen at 6% combined and 1% reflection, so the early return never triggers, and the live _previous* slots equal the current values. With a live owner this would be Medium: exempt wallets charged, and fee changes silently undone by the next exempt transfer.

RECOMMENDATION

No action on the live contract. Future versions: compute takeFee into local fee values instead of mutating storage, or include _burnFee in the early-return check and update _previous* in every setter.

SMC AUDITS

SMC-I-03 INFO Open

Swap-back reverts for valid fee configurations

CATEGORY

Denial of service; arithmetic

LOCATION

swapTokens(uint256) L1734-1735 and L1759-1763, addLiquidity L1811-1824, _transfer L1566-1575

DESCRIPTION

swapTokens divides by _combinedLiquidityFee (division by zero if marketing, developer, charity and LP fees are all 0) and always calls addLiquidity(otherLiquidityHalf, liquidityBalance), which the Pancake router rejects when either amount is 0 (INSUFFICIENT_AMOUNT), for example when _liquidityPoolFee is 0. A threshold of 0 makes the router pull 0 tokens, which _transfer rejects. In all three cases every sell that triggers the swap-back reverts.

IMPACT

Not reachable today (LP fee 1, combined 6, threshold 20M, all frozen). With a live owner each of these ordinary-looking settings would halt sells. Reflection-only or no-LP configurations of this template are unsafe.

RECOMMENDATION

No action on the live contract. Future versions: skip the liquidity step when its share is 0, guard the division, and enforce a non-zero threshold.

SMC AUDITS

SMC-I-04 INFO Open

Fee-on-transfer on every transfer breaks standard integrations

CATEGORY

Fee, tax, and reflection mechanics; integration safety

LOCATION

_transfer L1540-1584, _transferStandard L1603-1638

DESCRIPTION

The 7% tax applies to buys, sells and wallet-to-wallet transfers unless one side is exempt. Contracts that assume received == amount break. Pancake removeLiquidityETH sends the tokens pair to router to user and forwards the pre-tax amount, so it always reverts for TWD; the SupportingFeeOnTransferTokens variant works but charges the tax twice.

IMPACT

No loss inside the token. Integrators and LPs lose value or hit reverts: an LP who added 100,000,000 TWD (93,000,000 reached the pool) got back 80,568,929 TWD through the supporting remove path.

RECOMMENDATION

Holders and integrators: use the fee-on-transfer-supporting router functions, measure balance deltas, and remove liquidity to WBNB with removeLiquidity to pay the tax once. Future versions: limit taxes to pair trades, or document them prominently.

SMC AUDITS

SMC-I-05 INFO Open

Reflection side effects: skimmable pair balance, stranded reflections, rounding

CATEGORY

Fee, tax, and reflection mechanics; arithmetic and precision

LOCATION

_reflectFee L1668-1674, _getCompleteTaxValue L1658-1666, feeTransfer L1690-1703, excludeAccountFromReward L1518-1525

DESCRIPTION

No address is excluded from reflections, including the pair, 0x...dEaD (32.6% of supply) and the lockers. The pair's balance grows above its reserve on every taxed transfer, and anyone can skim the excess. Lockers return only the locked amount, so reflections credited to them are stranded. The total tax is floored once while its three parts are floored separately, so up to 2 wei per transfer becomes extra reflection.

IMPACT

Small, and no principal is lost. After a 1,000,000,000 TWD wallet transfer the pair held 1,499,496 TWD above its reserve, and a skimmer received 1,394,531 TWD net. PinkLock02 holds 31,045,116,772 TWD against a 30,000,000,000 lock, so about 1.045B TWD stays behind after unlock. About a third of every reflection goes to 0x...dEaD.

RECOMMENDATION

No action possible on the live contract (reward exclusion is onlyOwner). Future versions: exclude the pair, the dead address and lockers from reflections at deployment.

SMC-I-06 INFO Open

BNB accumulates in the token contract with no way out

CATEGORY

Token and supply logic; external call safety

LOCATION

receive() L1344, swapTokens L1733 and L1741, addLiquidity L1811-1824

DESCRIPTION

swapTokens only distributes address(this).balance - initialBalance, and addLiquidity ignores the router's return values, so BNB refunded by addLiquidityETH and any BNB sent to receive() stay in the contract. No function sends it out.

IMPACT

0.2756 BNB is stuck today, and every swap-back adds a little more. Only the developer and liquidity shares are affected, not holders.

RECOMMENDATION

Nothing on the live contract. Do not send BNB to the token address. Future versions: distribute the full balance or add a bounded, event-emitting rescue function.

SMC AUDITS

SMC-I-07 INFO Open

Code quality and ERC-20 edge cases

CATEGORY

Best practice / code quality; standards compliance

LOCATION

pragma L3, _transfer L1547, transferFrom L627-644, swapETHForTokens L1792-1809, ERC20 _balances / _totalSupply / _mint / _burn L491-495 and L750-788, setters L1375-1465

DESCRIPTION

Floating pragma ^0.8.0, built with 0.8.15; none of that version's known compiler bugs applies to a live path. Zero-amount transfers revert, which some batch senders and routers do not expect. transferFrom runs the full transfer, including swap-back external calls, before checking the allowance; the revert is atomic, so nothing is lost. swapETHForTokens, the Address helpers and the base ERC-20 storage and mint/burn code are dead. No configuration setter emits an event.

IMPACT

No security impact on the live contract. It makes integrations and monitoring harder (the admin history had to be rebuilt from archive state).

RECOMMENDATION

Future versions: pin a current compiler, allow zero-amount transfers, check the allowance before transferring, delete dead code and emit events for configuration changes.

SMC AUDITS

SMC-I-08 INFO Open

Swap-back uses zero slippage limits

CATEGORY

Front-running and MEV exposure

LOCATION

swapTokensForEth L1772-1790, addLiquidity L1811-1824

DESCRIPTION

The swap-back sells with amountOutMin = 0, adds liquidity with minimums of 0, and uses block.timestamp as the deadline. Anyone can trigger it with a sell to the pair.

IMPACT

A classic sandwich does not work. A front-running sell made while the inventory is at or above the threshold runs the swap-back itself first. In the only workable ordering, the attacker loses about 13% of the sandwich size to the 7% tax charged twice. At most 18,333,333 TWD (0.0587 BNB) is exposed per cycle. Considered for Low and kept at Informational because of these measurements.

RECOMMENDATION

Nothing on the live contract. Future versions: use a TWAP- or reserve-based minimum output and a cap on the swap size.

09

09 Remediation status

After the project addresses the findings, SMC Audits re-audits the changes and records the outcome here. The reviewed contract is immutable (not a proxy) and ownership is renounced, so on-chain code changes are not possible. SMC-H-01 and SMC-M-01 can be reduced by the project team through actions outside the contract (key management commitments, relocking or burning the PinkLock tokens); the rest is advisory.

ID	SEVERITY	STATUS	VERIFIED FIX
SMC-H-01	HIGH	Open	None (immutable). Mitigation depends on the developer wallet key holder; monitor eth_getCode(0xd769...b003).
SMC-M-01	MEDIUM	Open	None. Project can relock or burn lock 1069189.
SMC-L-01	LOW	Open	None (immutable). In-contract mitigations (increaseAllowance / decreaseAllowance).

SMC AUDITS

SMC-I-01	INFO	Acknowledged	Mitigated: powers neutralized by renounce to 0x0; verified live and in full history.
SMC-I-02	INFO	Open	None. Latent; unreachable with the frozen configuration.
SMC-I-03	INFO	Open	None. Latent; unreachable with the frozen configuration.
SMC-I-04	INFO	Open	None. By design; integration guidance.
SMC-I-05	INFO	Open	None (immutable). Applies to future versions.
SMC-I-06	INFO	Open	None (immutable). 0.2756 BNB stuck.
SMC-I-07	INFO	Open	None. Applies to future versions.
SMC-I-08	INFO	Open	None. Measured as unprofitable to sandwich.

Open items for completeness: (1) The Mudra locker (0xae7e...1252) holding 85.46% of LP is not verified; its owner EOA can call setMigrator (currently unset), and whether migrate requires the lock owner could not be confirmed from source. This affects the LP-lock statement, not any finding's severity. (2) The EIP-7702 evidence for SMC-H-01 is a fork simulation of the post-authorization code, a live mainnet eth_call that runs delegated code, and a real sweeper delegation found on BSC. (3) The holder list is from GoPlus and on-chain balanceOf at the time of review.

10

10 Conclusion

At the reviewed address, TWD / THE WORD is an unmodified CoinManufactory reflection token with a fixed 7% tax, a PancakeSwap swap-back, and renounced ownership. The review found no critical issue.

The high and medium findings do not come from owner(), which is renounced and whose functions our live checks confirm are dead. They come from two privileges that the renounce does not remove: the permanent developer fee receiver, which can stop every sell on the main pair through EIP-7702 and restart it at will (SMC-H-01), and an expired lock that gives the deployer 30% of supply to sell fee-free (SMC-M-01). The remaining findings are one low-severity approve race and eight informational notes, and none of them puts user funds at risk in the deployed state.

The load-bearing condition is that the developer wallet 0xd76914B765625475D2524EbbC8503457CCEBb003 never holds code that rejects a 2300-gas BNB transfer. That condition is true today, but only that wallet's key holder can keep it true, and they can change it in one transaction. Readers should check eth_getCode on that address and the state of PinkLock lock 1069189 themselves, and should not read "ownership renounced" as "no admin risk".

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 commit 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