SMC AUDIT

SMART CONTRACT
SECURITY AUDIT



×

AUDIT REPORT

ZDK (ZEDEK) Token Security Review

PROJECT	ZDK / ZEDEK ("ZEDEK")
CHAIN	BNB Chain (BSC, chain id 56)
CONTRACT	0xCbEaaD74dcB3a4227D0E6e67302402E06c119271
VERSION	Verified source (Sourcify full match, matchId 7034137) · Solidity 0.8.17
REPORT ID	SMC-ZDK-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 ZDK / ZEDEK smart contract on BNB Chain at address 0xCbEaaD74dcB3a4227D0E6e67302402E06c119271, using the source verified on Sourcify (a full creation and runtime bytecode match, matchId 7034137).

The review ran from 28 September 2026 to 30 September 2026 and covered the deployed ZEDEK contract in full, its transfer, fee, burn, allowance, swap-and-liquify, and administrative logic, and its live on-chain state. The manual review, in three separate passes, was supported by cast and Foundry 1.8.3 fork tests against BNB Chain (round 1: 10 passed; round 2: 14 passed, including 256 fee-cap fuzz runs, 64 wallet-transfer fuzz runs, and 32 sell fuzz runs) and by Slither 0.11.6 and Aderyn 0.6.8, whose results were all triaged.

The review identified 11 findings: 0 critical, 1 high, 2 medium, 3 low, and 5 informational. No issue in the live configuration lets an arbitrary caller take another holder's balance, and a supporting Pancake buy and sell both succeeded on a fork. The most significant issues concerned the owner, a live EOA, which can block holder sells to the pair by removing the pair's limit exemption and setting the wallet cap below the pair's balance, then still sell after turning swap-and-liquify off (SMC-H-01, High); and the live tax inventory, which can be walked, by ordinary sells and with no owner call, into a raw-unit band where every later sell reverts until someone transfers tokens to the contract (SMC-M-02, Medium). That band is not the current inventory. The other live code-level findings are capped but immediate fee and wallet control (SMC-M-01, Medium), a division-by-zero sell revert if swap fees are cleared (SMC-L-01, Low), the ERC-20 approve race (SMC-L-02, Low), and zero-slippage swap-and-liquify (SMC-L-03, Low).

Overall, the code was found to be a standard taxed TokensByGen contract with fee caps, no mint, and a one-way trade-open switch, plus an owner exception in the wallet-limit check that becomes a sell switch. Its integrity is contingent on the owner key at 0x7A156c4CC939b029833bB06f4e43013f24DD8f90 remaining unused for that switch, a condition that is true in the reviewed state and reversible by the owner at any time, or removable by renounce.

0	1	2	3	5	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 not 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, integrators, and any future redeployment.

03

03 Project overview

ZDK (ZEDEK, on-chain name "ZEDEK") is a taxed BEP-20 token on BNB Chain. This audit covers the single deployed ZEDEK contract that implements balances, allowances, buy, sell, and wallet-transfer fees, an optional deflationary burn, PancakeSwap V2 swap-and-liquify, and the owner's parameter controls. The source identifies itself as a TokensByGen template compiled with solc 0.8.17. It is not a proxy and is not upgradeable.

System at a glance

COMPONENT	ROLE
ZEDEK token contract	Balances, transfer / transferFrom, allowances, and the fee split. Initial supply 700,000,000,000,000 tokens at 9 decimals. Live totalSupply is 663,490,846,148,244. There is no mint function.
Fee, burn, and swap-and-liquify	Live buy fee 3% (2% marketing + 1% dev), all accumulated on the contract for later swap. Live sell fee 4% (1% marketing + 1% liquidity + 1% burn + 1% dev). Burn reduces totalSupply because deflationaryBurn is true. Wallet-to-wallet transfers use the buy fee. Caps in code: 10% buy and 15% sell, including the 1% dev fee.
Admin / ownership	owner() is the EOA 0x7A156c4CC939b029833bB06f4e43013f24DD8f90. It can set fees inside the caps, set max transaction and max wallet down to 0.1% and 0.5% of initial supply, exempt any address from fees or limits, redirect the three project wallets, toggle swap-and-liquify, clear the dev fee by paying 3 BNB, and renounce. It cannot mint, blacklist, or close trading once TradeOpen is true. Trading is open. The owner is not fee-exempt, limit-exempt, or whitelisted.
External integrations	Router 0x10ED43C718714eb63d5aA57B78B54704E256024E (PancakeSwap V2). Pair 0x246d1711a3834c405845ae52dE0B808EF9BFba6E holds about 121.65 trillion ZDK and about 54.3 BNB. Dev fee BNB is sent to feeCollector 0x3F679d42922CA9187694C33f4c86E87a83515420 (verified FeeCollector). Auto-LP is sent to Wallet_Liquidity. The pair and locker are out of scope beyond the calls this token makes.

Key mechanisms to note

- Every taxed transfer pays the live fee. A non-exempt wallet transfer of 1,000,000 tokens delivers 970,000 to the recipient and 30,000 to the contract. A sell delivers 96% to the pair, sends 3% to the contract, and burns 1%.
- Ownership has not been renounced, and the wallet-limit check treats the owner differently from everyone else. The live owner can remove the pair's limit exemption and set the wallet cap below the pair's balance. Holder sells then revert with E15, while from != owner() lets the owner skip that check. Renounce would send _owner to address(0) and make every onlyOwner call fail. That has not happened. This is the load-bearing fact in the current security posture.
- The on-chain strings are not an on-chain lock. PinkLock getLockById(1073649) returns owner address(0), lock 2023-02-19, unlock 2169-05-19, about 95.09% of the Cake-LP, and unlock requires

owner == msg.sender, so that LP cannot be withdrawn. The project wallet 0xf4D0...8794 holds about 4.91%. New auto-LP is sent to that project wallet, not to the locker.

04

04 Scope

ITEM	DETAIL
Project	ZDK / ZEDEK ("ZEDEK")
Chain	BNB Chain (BSC, chain id 56)
Language	Solidity
Contract	0xCbEaaD74dcB3a4227D0E6e67302402E06c119271 (verified source via Sourcify)
Commit / version	Deployed bytecode; Sourcify full match (matchId 7034137, creation and runtime both match, verified 2025-07-10). Not a proxy. Deployed at block 25253703.
Compiler	solc 0.8.17+commit.8df45f5f; optimizer enabled, runs 200; EVM version london. Pragma 0.8.17 (fixed).

Files reviewed

FILE	LINES	NOTES
ZEDEK.sol	1193	Entire flattened file: IERC20, SafeMath, Context, Address, Uniswap V2 factory/pair/router interfaces, ZEDEK. SHA-256 b1ca2a8f...a83d1c3.
FeeCollector.sol	49	Sourcify-verified collector at 0x3F67...5420. Reviewed for control of its BNB and for any call back into ZEDEK.

Out of scope

Off-chain components, the project website and front end, and the Telegram channel named in the on-chain strings. Liquidity pools, DEX pairs, routers, and market or economic conditions, except where this contract calls the router or reads the pair it created. PinkLock record 1073649 was decoded (owner address(0), unlock 2169-05-19, unreachable). Deployment process and key management history. Exemption setters emit no event, so those calls cannot be recovered from logs; their absence does not affect SMC-H-01 or SMC-M-02, both executed against the deployed bytecode on a mainnet fork.

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	Owner can block holder sells to the Pancake pair	HIGH	Open
SMC-M-01	Owner can raise fees to the cap and redirect fee wallets	MEDIUM	Open
SMC-M-02	Small tax inventory makes every Pancake sell revert	MEDIUM	Open
SMC-L-01	swapAndLiquify divides by zero when swap fees are cleared	LOW	Open
SMC-L-02	ERC-20 approve race condition (SWC-114)	LOW	Open
SMC-L-03	Swap-and-liquify accepts any price	LOW	Open
SMC-I-01	Fee-on-transfer breaks naive integrations	INFO	Open
SMC-I-02	decimals declared as uint256 instead of uint8	INFO	Open
SMC-I-03	Process-now percent cap is a no-op comparison	INFO	Open
SMC-I-04	Failed BNB transfers are ignored	INFO	Open
SMC-I-05	Ownership transfer drops exemptions and does not grant them	INFO	Open

0	1	2	3	5	11
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

SMC AUDITS

08

08 Detailed findings

SMC-H-01 HIGH Open

Owner can block holder sells to the Pancake pair

CATEGORY

Pause, blacklist, and emergency switches

LOCATION

_transfer(address,address,uint256) L924-928, Wallet_Settings__Exempt_From_Limits L791-798, Maintenance__New_Liquidity_Pair L625-633, Contract_SetUp_4__Set_Limits L582-600, Processing__Auto_Process L729-732

DESCRIPTION

The wallet-cap check is `if (!_isLimitExempt[to] && from != owner())` and then `require((heldTokens + amount) <= max_Hold, "E15")`. The pair is limit-exempt today, and max_Hold is 140 trillion tokens (20% of initial supply), above the pair's balance of about 121.65 trillion, so sells succeed. Both conditions are owner-writable.
Wallet_Settings__Exempt_From_Limits(pair, false) clears the exemption.
Maintenance__New_Liquidity_Pair(pair, false) clears _isPair and _isLimitExempt together.
Contract_SetUp_4__Set_Limits can set the wallet cap as low as 3.5 trillion tokens, and the pair already holds more. After the exemption is cleared and the cap is at the floor, every holder transfer to the pair reverts with E15. The same check is skipped when from == owner(), so the owner's own transfer to the pair is not subject to the cap. The automatic swap uses the contract as from, so it also reverts until the owner calls Processing__Auto_Process(false). There is no blacklist and no way to set TradeOpen back to false.

IMPACT

Holders would be unable to sell on the pool that this contract created, while the owner could disable the swap and sell in chunks up to the transaction cap. Buys and small wallet-to-wallet transfers still succeed, so this is not a full freeze of every transfer and it does not move balances by itself, hence High rather than Critical. It is High rather than Medium because the owner is a live EOA, the two calls have no timelock, and the pair balance is already above the lowest legal wallet cap. If ownership were renounced, the same code would be Informational; the fork test test_renounce_neutralizes_owner_powers shows renounce blocks the exemption call. Renounce has not been called.

PROOF OF CONCEPT

On a BNB Chain fork, test_H01_owner_can_block_holder_sells (ZdkFork.t.sol) has a holder sell successfully first. The owner then clears the pair exemption and sets both limits to their floors. A transfer of 1 token to the pair reverts E15, and the supporting router sell reverts. A later 0.01 BNB buy still succeeds. After Processing__Auto_Process(false), the owner sells 10 tokens and receives BNB. The control test test_live_state_and_amm_round_trip confirms the live configuration is not yet armed (a 0.2 BNB buy returned 97% of quote and the following sell returned BNB). Round 2 added test_H01_second_path_new_liquidity_pair and the gross-versus-net edge; both passed. The full tests are provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. The owner should renounce, or move the role to a timelocked multisig that will not call these two functions. Holders should treat the owner key as

able to disable Pancake sells. Future versions should keep registered pairs permanently limit-exempt, delete the from != owner() exception, and enforce the wallet cap only on non-pair recipients.

SMC-M-01 MEDIUM Open

Owner can raise fees to the cap and redirect fee wallets

CATEGORY

Privilege and centralization risk

LOCATION

Contract_SetUp_2__Fees_on_Buy L518-547, Contract_SetUp_3__Fees_on_Sell L550-579, Update_Project_Wallets L683-700, addLiquidity L1063-1074, Wallet_Settings__Exclude_From_Fees L801-809

DESCRIPTION

Fee setters revert above 10% on buys (E01) and 15% on sells (E02), and both limits include the current _Fee__Dev of 1. Inside those caps the owner can change rates immediately. Update_Project_Wallets moves Wallet_Marketing, Wallet_Liquidity, and Wallet_Tokens to any non-zero addresses. Swap-and-liquify BNB, after the dev slice, is sent to Wallet_Marketing, and Cake-LP from addLiquidity is minted to Wallet_Liquidity. Those three wallets are currently the same EOA, which is not fee-exempt. The owner can also exclude any address from fees. There is no mint function.

IMPACT

A fee move to the cap takes 15% of a sell and 10% of a buy or wallet transfer, which is material and can be front-run ahead of a large trade, but it cannot seize the remaining balance. Redirecting the project wallets redirects future tax BNB and auto-LP, not the tokens already in user wallets and not the LP in PinkLock. That is meaningful and limited by the caps, hence Medium. The sell-blocking combination is SMC-H-01, not this finding. If the owner renounced, this finding would be Informational.

PROOF OF CONCEPT

test_M01_fee_and_limit_bounds: the owner sets a 10% buy (accepted) and an 11% buy (reverts E01), a 15% sell (accepted) and a 16% sell (reverts E02). Limits below the 0.1% / 0.5% floors revert E03 and E04. Opening trade again reverts E05. With the sell fee at the cap, a 10,000 token transfer to the pair credited the pair 8,500 tokens and reduced totalSupply by 100. The full test is provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. Holders should assume fees can move to 10% and 15% without notice, and that new auto-LP accrues to an EOA the owner can change. Future versions should use a timelocked multisig, basis-point fees with a lower ceiling, and a fixed liquidity recipient that cannot be swapped for an EOA after launch.

SMC-M-02 MEDIUM Open

Small tax inventory makes every Pancake sell revert

CATEGORY

Denial of service

LOCATION

swapAndLiquify(uint256) L991-1041, swapTokensForBNB(uint256) L1045-1058, _transfer L946-970

DESCRIPTION

A sell to the pair calls swapAndLiquify when swapCounter >= swapTrigger, the swap is enabled, and the contract holds tokens. The swap size is the whole contract balance. swapTokensForBNB uses amountOutMin 0; Pancake reverts with INSUFFICIENT_OUTPUT_AMOUNT when the quoted WBNB out is 0. When the quoted out is positive but the liquidity leg rounds to 0 while LP_Tokens > 0, addLiquidityETH is called with 0 BNB and reverts. On the live reserves, processing reverts for every power-of-two inventory from 8 through 16,384 raw units and succeeds at 32,768 raw units. The owner is not required to enter the band: the counter can be advanced with small wallet transfers, then four triggered sells walk the inventory from 858 million tokens down to 1,500 raw units, after which the next sell to the pair reverts. That revert does not change storage, so the counter stays at the trigger and every later sell reverts. Buys do not take this path; wallet transfers still succeed.

IMPACT

While the inventory sits in the band, Pancake sells revert for every holder, including the owner, until the inventory is topped up or the owner disables the swap. No balance is taken. The live inventory is not in the band, a normal sell was shown to succeed, and the top-up is permissionless, so this is not High. It is not Low: reaching the band does not require the owner, the halted function is the sell, and the halt sticks until the top-up.

PROOF OF CONCEPT

Fork tests test_permissionless_shrink_can_brick_next_sell, test_dust_window_reverts_and_donation_clears, and test_organic_process_remainder_trajectory (ZdkRound2.t.sol) all pass. The first logs the inventory after each triggered sell (858228245370250036, 178336181249002, 37153349175, 7738790, 1500) and ends with the next sell reverting. Transferring 1,000,000 tokens to the contract clears the halt and the next sell succeeds. The full tests are provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. Before swapping, skip the call when the quoted WBNB out is 0 or when the BNB liquidity leg rounds to 0. Holders who see sells revert while buys still work can transfer a small amount of ZDK to the token contract and retry. Future versions should not let a reverting swap leave swapCounter at the trigger.

SMC-L-01 LOW Open

swapAndLiquify divides by zero when swap fees are cleared

CATEGORY

Arithmetic, overflow, and precision

LOCATION

swapAndLiquify(uint256) L999-1000, Maintenance__Remove_Contract_Fee() L668-679, Contract_SetUp_2 L544, Contract_SetUp_3 L576, Processing__Update_Swap_Trigger_Count L746-749

DESCRIPTION

LP_Tokens is computed as Tokens * (_Fee__Buy_Liquidity + _Fee__Sell_Liquidity) / _FeesTotal / 2, where _FeesTotal is _SwapFeeTotal_Buy + _SwapFeeTotal_Sell. Those swap totals are 3 and 3 today, so the division is safe. They become zero only if the owner first pays exactly 3 BNB to clear _Fee__Dev and then sets buy and sell marketing and liquidity to zero. The next transfer to the pair then enters swapAndLiquify and hits a Solidity 0.8 division-by-zero panic (0x12). Wallet-to-wallet transfers do not enter that function.

IMPACT

Sells to the pair would revert for as long as the contract holds tokens, swap-and-liquify stays enabled, and the trigger condition stays true. Peer transfers would still work. Reaching that state costs 3 BNB and several owner calls, and it is not the live fee state, hence Low.

PROOF OF CONCEPT

test_L01_zero_swap_fee_reverts_sells. A 1,000 token transfer to the pair succeeds under live fees. The owner then pays 3 BNB, zeroes both fee setters, and sets the swap trigger count to 0. A peer transfer of 1,000 tokens succeeds, and the next transfer to the pair reverts with Panic(0x12). The test passes. The full test is provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. The owner should not clear swap fees while the contract holds ZDK and swap-and-liquify is on. Future versions should return early when _FeesTotal == 0 instead of dividing.

SMC AUDITS

SMC-L-02 LOW Open

ERC-20 approve race condition (SWC-114)

CATEGORY

Approvals and allowance handling

LOCATION

approve(address,uint256) L865-876, transferFrom L883-887, increaseAllowance L855-858, decreaseAllowance L860-863

DESCRIPTION

approve writes _allowances[owner][spender] = amount with no reference to the previous allowance. transferFrom later subtracts the full amount from that allowance. A spender who sees an owner transaction lowering the allowance from 100 to 50 can transfer 100 first and 50 after the new approval, for 150. increaseAllowance and decreaseAllowance are present and do not have this shape.

IMPACT

The spender can move more tokens than the owner's second approval was meant to allow. It requires a prior allowance, a new approve that is not zero-to-N, and transaction ordering, hence Low. It is not available against a wallet that never approves.

PROOF OF CONCEPT

test_L02_approve_race, using the project wallet's live ZDK balance. The owner approves 100, the spender transfers 100, the owner approves 50, and the spender transfers 50. The owner wallet is debited 150 tokens and the allowance ends at 0. The full test is provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. Holders and routers should use increaseAllowance and decreaseAllowance, or set the allowance to zero and wait for that transaction before setting a new value. Future versions should rely on those incremental functions or use ERC-20 Permit.

SMC AUDITS

SMC-L-03 LOW Open

Swap-and-liquify accepts any price

CATEGORY

Front-running and MEV exposure

LOCATION

swapTokensForBNB(uint256) L1045-1058, addLiquidity(uint256,uint256) L1063-1074

DESCRIPTION

swapExactTokensForETHSupportingFeeOnTransferTokens is called with amountOutMin 0 and deadline block.timestamp. addLiquidityETH is called with amountTokenMin 0 and amountETHMin 0. A searcher can sell ZDK immediately before this swap and buy it back after. The tokens being sold are the tax balance on the contract, not a user's wallet. Today that balance is about 1.6 billion tokens, far below the pair's 121.55 trillion, and swapCounter (9) is still under swapTrigger (11), so the next ordinary sell does not swap yet. max_Tran is larger than the pair's ZDK reserve, so a much larger contract balance would be allowed to swap.

IMPACT

The loss is value in the tax inventory and in the pool's BNB side, bounded by how many tokens the contract swaps. At the current balance the extractable amount is small, and a searcher cannot force an arbitrary user to donate it, so the rating is Low. It would matter more if the contract were allowed to accumulate up to max_Tran before swapping.

PROOF OF CONCEPT

test_L03_zero_slippage_accepts_dumped_price. Against the unmanipulated pool, Processing__Process_Now(100) reduced the pair's WBNB reserve by 3.09 BNB. After an attacker dumped ZDK first, the same processing reduced the reserve by 0.27 BNB. Both calls succeeded. The full test is provided in the engagement materials.

RECOMMENDATION

The live bytecode cannot be patched. Future versions should pass a minimum BNB out and a minimum LP amount computed from reserves with a small tolerance, and cap the swap at a fraction of the pair's reserve rather than at max_Tran alone.

SMC AUDITS

SMC-I-01 INFO Open

Fee-on-transfer breaks naive integrations

CATEGORY

Fee, tax, and reflection mechanics

LOCATION

_tokenTransfer L1092-1177, transfer L878-881, transferFrom L883-887

DESCRIPTION

When a fee is taken, the sender's balance falls by the full tAmount, the recipient is credited the net, and only that net is emitted in Transfer. The burn, the contract's swap inventory, and Wallet_Tokens are updated with no Transfer event. swapExactTokensForETH on the Pancake router assumes the pair receives amountIn; it does not, and that call reverts. The supporting sell function measures the pair's balance delta and succeeds.

IMPACT

Indexers that sum Transfer amounts will not match balanceOf. A router, bridge, or pair that does not support fee-on-transfer will revert on sells or credit the wrong amount on buys. Users on the supporting Pancake path are not stuck. This is an integration hazard of a fee token working as written, hence Informational.

RECOMMENDATION

The live bytecode cannot be patched. Integrators must use the fee-supporting router methods and must not reconstruct balances from Transfer events alone. Future versions should emit a Transfer for each fee leg.

SMC-I-02 INFO Open

decimals declared as uint256 instead of uint8

CATEGORY

Standards compliance

LOCATION

decimals() L839-841

DESCRIPTION

decimals is stored and returned as uint256 with value 9. The BEP-20 / ERC-20 decimals function is specified to return uint8. The ABI word is still 32 bytes, so a caller that ABI-decodes a uint8 receives 9. The verified Sourcify ABI records the return type as uint256.

IMPACT

Strict ABI checkers and some typed clients reject or mis-type the function. Runtime decoding still yields 9. There is no balance or fee impact, hence Informational.

RECOMMENDATION

The live bytecode cannot be patched. Integrators should read the verified ABI rather than assume uint8. Future versions should declare uint8 public constant decimals = 9.

SMC AUDITS

SMC-I-03 INFO Open

Process-now percent cap is a no-op comparison

CATEGORY

Arithmetic, overflow, and precision

LOCATION

Processing__Process_Now(uint256) L736-742

DESCRIPTION

The function tries to clamp inputs above 100 with `if (Percent_of_Tokens_to_Process > 100) {Percent_of_Tokens_to_Process == 100;}`. The == is a comparison and does not assign. sendTokens is then balance * percent / 100. At 101 this exceeds the balance, swapAndLiquify hits E13 ("Not enough tokens"), and the transaction reverts. At 100 the swap runs. The call is onlyOwner.

IMPACT

The owner cannot accidentally over-swap: values above 100 revert instead of clamping, which is safer than the comment, and it does not affect holders, hence Informational. It will surprise an operator who passes 101 expecting a full process.

RECOMMENDATION

The live bytecode cannot be patched. The owner should pass a percent from 1 to 100. Future versions should assign percent = 100 when the input is higher, or reject it with a clear error.

SMC AUDITS

SMC-I-04 INFO Open

Failed BNB transfers are ignored

CATEGORY

Reentrancy and external call safety

LOCATION

send_BNB(address,uint256) L890-894, swapAndLiquify(uint256) L1022-1032

DESCRIPTION

send_BNB performs payable(_to).call{value: _amount}("") and returns the success flag. Callers ignore it. In swapAndLiquify the dev slice is sent to feeCollector and the remainder to Wallet_Marketing. If the marketing wallet reverts, the BNB stays on the token contract and the swap still finishes. inSwapAndLiquify is set before the external calls and cleared after, so a receiver cannot re-enter swapAndLiquify. The live marketing wallet is an EOA.

IMPACT

A reverting marketing wallet strands that swap's BNB on the token until a later swap sends to a wallet that accepts BNB. Holders' ZDK sells are not reverted by this path. The live wallet accepts BNB, hence Informational. If the owner renounced while the marketing wallet rejected BNB, the stuck BNB would have no recovery path.

RECOMMENDATION

The live bytecode cannot be patched. The owner should keep Wallet_Marketing as an EOA or a contract that accepts BNB. Future versions should revert when send_BNB returns false, and include a separate BNB recovery path.

SMC AUDITS

SMC-I-05 INFO Open

Ownership transfer drops exemptions and does not grant them

CATEGORY

Privilege and centralization risk

LOCATION

Maintenance__Ownership_Transfer(address) L637-652

DESCRIPTION

The transfer function clears _isLimitExempt, _isExcludedFromFee, and _isWhiteListed on the current owner, emits OwnershipTransferred, and sets _owner to newOwner. It rejects address(0) (E06). It does not set any of those flags on the new owner. The new owner can call the setters, including the one that exempts themselves. Renounce is a separate function and does go to address(0). The live owner is already not fee-exempt, not limit-exempt, and not whitelisted.

IMPACT

A new owner who does not notice will pay fees and be subject to the caps. They do not lose the ability to call onlyOwner, so they are not locked out. Holders are not trapped by the transfer itself. The SMC-H-01 bypass from == owner() follows the new owner immediately. This is a footgun for the incoming owner, not a holder-facing bug, hence Informational.

RECOMMENDATION

Future versions should set the three flags on the new owner inside the same transaction, and emit them. An incoming owner should call the exemption setters before moving size.

SMC AUDITS

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 not a proxy, so the bytecode cannot be changed in place. Ownership has not been renounced, so the owner can still change fees, limits, exemptions, swap settings, and the project wallets. Renounce would neutralize SMC-H-01 and SMC-M-01; it would not repair SMC-M-02, SMC-L-02, SMC-L-03, or the informational items. SMC-M-02 is cleared, while it is happening, by transferring tokens to the contract or by Processing__Auto_Process(false).

ID	SEVERITY	STATUS	VERIFIED FIX
SMC-H-01	HIGH	Open	None. Not armed: the pair is still limit-exempt and the wallet cap is 20%. Renounce would close it.
SMC-M-01	MEDIUM	Open	None. Caps hold at 10% and 15%. Owner can still move rates and wallets.

SMC-M-02	MEDIUM	Open	None. Not armed: inventory is about 858 million tokens. A transfer to the token clears the band.
SMC-L-01	LOW	Open	None. Live swap-fee totals are 3 and 3. Arming it requires the 3 BNB dev-fee removal.
SMC-L-02	LOW	Open	None (immutable). increaseAllowance / decreaseAllowance are already present.
SMC-L-03	LOW	Open	None (immutable). Current contract inventory is small relative to the pair.
SMC-I-01	INFO	Open	None (immutable). Supporting router sells succeed.
SMC-I-02	INFO	Open	None (immutable). Value is 9; the ABI type is uint256.
SMC-I-03	INFO	Open	None (immutable). Inputs above 100 revert.
SMC-I-04	INFO	Open	None. Live marketing wallet is an EOA and accepts BNB.
SMC-I-05	INFO	Open	None. Live owner already has no exemption flags.

SMC AUDITS

10

10 Conclusion

At the reviewed address, ZDK / ZEDEK is a TokensByGen taxed BEP-20 with fee caps, no mint, deflationary sell-burn currently enabled, and trading already open. The review found no critical severity issue, and one high severity issue, in the live contract.

Holder sells work in the current configuration, which the fork round trip confirms. They remain available only while the owner at 0x7A156c4CC939b029833bB06f4e43013f24DD8f90 leaves the pair limit-exempt and leaves the wallet cap above the pair's balance, and while the contract's token inventory stays above the band in SMC-M-02. The remaining findings are two medium items, three low items, and five informational notes. None of them, in the deployed state, let an arbitrary caller drain another wallet.

The load-bearing conditions are that this owner key is not used to clear the pair's limit exemption and tighten the wallet cap, that it is not compromised, and that sells are retried or the inventory is topped up if SMC-M-02 is reached. Readers should re-check owner(), _isLimitExempt on the pair, Token_Information, and the token's own balance before relying on the token. The LP in PinkLock record

1073649 is a separate fact: the lock owner is address(0) and the unlock date is 2169-05-19, so that portion cannot be pulled. New auto-LP is sent to the project EOA, which still holds about 4.91%.

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