SMC AUDIT

SMART CONTRACT
SECURITY AUDIT



AUDIT REPORT

GIANT (GTAN) Token Security Review

PROJECT	GIANT (GTAN)
CHAIN	BNB Chain (chainId 56)
CONTRACT	0xbD7909318b9Ca4ff140B840F69bB310a785d1095
VERSION	Verified source (Sourcify / BscScan) · Solidity 0.8.19
REPORT ID	SMC-GIANT-202610-01
DATE	5 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 GIANT (GTAN) token contract on BNB Chain at its verified source (Solidity 0.8.19). The review covered the token, its transfer and fee logic, the swap-and-liquify fee processing, the holding and transaction limits, and the owner privileges.

The review identified 9 findings: 0 critical, 0 high, 2 medium, 2 low, and 5 informational. The most significant issue is an owner-triggerable condition that permanently blocks all sells, which becomes a one-way honeypot if ownership is renounced in that state. The remaining risk is standard owner centralization plus minor MEV and griefing exposure. No unprivileged path to theft, drain, or freeze was found.

Overall, the code is a constrained, competently built fee-token template: fees are capped, holding and transaction limits are floored, trade-open is a one-way switch, and there is no mint function and no blacklist.

0	0	2	2	5	9
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

As of this report, 0 of 9 findings have been resolved.

SMC AUDITS

03

03 Project overview

GIANT (GTAN) is a deflationary fee-on-transfer BEP-20 token on BNB Chain, built from the TokensByGen template. This audit covers the single token contract that implements the token, its transfer and fee logic, the swap-and-liquify fee processing, and its supply mechanics.

System at a glance

COMPONENT	ROLE
Token contract	BEP-20 logic, balances, transfers, 9 decimals
Fee module	Buy and sell fees (marketing, liquidity, burn, tokens, dev), capped at 10% buy / 15% sell
Swap-and-liquify	Converts accrued fee tokens to BNB and adds liquidity on PancakeSwap

Admin / ownership	Single owner (not renounced) sets fees, wallets, limits, exemptions, whitelist, and opens trade
External integrations	PancakeSwap V2 router and the GTAN/WBNB pair, both set once in the constructor and immutable

Key mechanisms to note

- Deflationary burn is active: a sell burn and a direct-to-burn transfer reduce total supply.
- Buy and sell fees are capped at set time (buy 10%, sell 15%, including a fixed dev fee).
- Holding and transaction limits have floors (transaction 0.1%, wallet 0.5% of initial supply).
- Trade-open is a one-way switch and cannot be re-paused once opened.
- The router and pair are fixed in the constructor with no setter.

04

04 Scope

ITEM	DETAIL
Project	GIANT (GTAN)
Chain	BNB Chain (chainId 56)
Language	Solidity
Contract	0xbD7909318b9Ca4ff140B840F69bB310a785d1095
Version	Verified source, Solidity 0.8.19
Compiler	solc 0.8.19 (default optimizer)

Files reviewed

FILE	LINES	NOTES
GIANT.sol	~1,196	token, fee logic, swap-and-liquify, limits, owner controls

Out of scope

Third-party dependencies (PancakeSwap router and pair, WBNB), off-chain components, the project front end and website, deployment and key management, initial liquidity provisioning and lock arrangements, and market or economic conditions.

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.
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 and owner checks, approvals and allowance handling, privilege and centralization risk, oracle and price handling, arithmetic and precision, upgradeability and proxy risk, reentrancy and external-call safety, pause/blacklist/emergency switches, token and supply logic, front-running and MEV exposure, fee/tax/reflection mechanics, 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-M-01	Zeroed swap fees permanently brick all sells	MEDIUM	Open
SMC-M-02	Owner centralization: mutable fees and owner-held auto-liquidity	MEDIUM	Open
SMC-L-01	Fee swap executed with zero slippage protection	LOW	Open
SMC-L-02	Max-wallet limit enables dust griefing	LOW	Open
SMC-I-01	Broken input clamp in Processing__Process_Now	INFO	Open
SMC-I-02	decimals() returns uint256 instead of uint8	INFO	Open
SMC-I-03	SwapAndLiquify event reports the wrong swapped amount	INFO	Open
SMC-I-04	Dev fee removed even if the 3 BNB payment fails to arrive	INFO	Open
SMC-I-05	Ownership transfer drops the new owner's exemptions	INFO	Open

0	0	2	2	5	9
CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL

SMC AUDITS

08

08 Detailed findings

SMC-M-01 MEDIUM Open

Zeroed swap fees permanently brick all sells

CATEGORY

Fee / swap-and-liquify logic

LOCATION

GIANT.sol : swapAndLiquify (fee-split division)

DESCRIPTION

swapAndLiquify computes `LP_Tokens = Tokens * (_Fee__Buy_Liquidity + _Fee__Sell_Liquidity) / _FeesTotal / 2`, where `_FeesTotal = _SwapFeeTotal_Buy + _SwapFeeTotal_Sell`. If the owner removes the dev fee with `Maintenance__Remove_Contract_Fee` and sets all marketing and liquidity fees to zero, `_FeesTotal` becomes zero. Fee tokens accrued under earlier non-zero fees stay on the contract, and `swapCounter` remains armed from prior buys. The next sell triggers swapAndLiquify and reverts with a division-by-zero panic.

IMPACT

Every sell reverts while buys still succeed, a one-way honeypot. It is recoverable while the owner can re-add a non-zero fee, but it becomes permanent if ownership is renounced in the zero-fee state, trapping all holder funds. The trigger is owner-only; an unprivileged actor cannot reach it.

PROOF OF CONCEPT

On a BNB Chain fork, after the owner removes the dev fee and zeros all fees, a holder's sell reverts with Panic(0x12). The state was reached both directly and organically: real buys arm the counter and accrue fee tokens under live fees, then the owner zeros all fees, and the next sell reverts. A redeploy of the patched contract on the same fork confirms the fix resolves it. A full Foundry fork test is provided in the engagement materials.

RECOMMENDATION

Guard the zero-fee case before the division:

```
uint256 _FeesTotal = (_SwapFeeTotal_Buy + _SwapFeeTotal_Sell);
if (_FeesTotal == 0) { inSwapAndLiquify = false; return; }
```

SMC-M-02 MEDIUM Open

Owner centralization: mutable fees and owner-held auto-liquidity

CATEGORY

Centralization

LOCATION

GIANT.sol : Contract_SetUp_2__Fees_on_Buy, Contract_SetUp_3__Fees_on_Sell, addLiquidity, Maintenance__Ownership_*

DESCRIPTION

Ownership is not renounced. The owner can change buy and sell fees at any time up to the caps (buy 10%, sell 15%), redirect the fee wallets, and mark arbitrary addresses as pairs. Auto-generated liquidity from swap-and-liquify is sent to an owner-controlled wallet; the LP lock is referenced only by an off-chain URL field and is not enforced on-chain.

IMPACT

The owner can raise the sell fee to the 15% cap after holders have bought in (a bounded soft-rug), and can move or withdraw auto-generated LP. The abuse is bounded by the fee caps, the limit floors, the one-way trade switch, the immutable router and pair, and the absence of any mint or blacklist. Even used in good faith, an un-renounced owner key is a single point of failure.

PROOF OF CONCEPT

All fee and wallet setters carry onlyOwner; the auto-LP destination is `Wallet_Liquidity`, an owner-set address. The on-chain owner is a live externally owned account; the LP-locker field is a display string with no on-chain effect. Confirmed from source and on-chain state.

RECOMMENDATION

Renounce ownership, or place it behind a timelock or multisig. Route auto-generated LP to a timelocked locker or a burn address. Document the owner capabilities clearly for holders.

SMC AUDITS

SMC-L-01 LOW Open

Fee swap executed with zero slippage protection

CATEGORY

MEV

LOCATION

GIANT.sol : swapTokensForBNB, addLiquidity

DESCRIPTION

The automated fee conversion calls `swapExactTokensForETHSupportingFeeOnTransferTokens` with `amountOutMin = 0` and `addLiquidityETH` with both minimums 0. Any taxed sell past the swap-trigger count runs this swap inside the seller's transaction.

IMPACT

A searcher can sandwich the contract's own fee swap. The value at risk is the project's fee BNB (marketing, liquidity, dev), not holder balances, and it is bounded by the fee tokens swapped per trigger. On the live pool this is about 48 ppm of the GTAN/WBNB reserve per trigger, so the extractable amount is negligible relative to gas.

PROOF OF CONCEPT

The swap passes `amountOutMin = 0`. A fork measurement shows the per-trigger swap is about 48 ppm of the pool reserve, bounding any sandwich to a trivial amount.

RECOMMENDATION

Derive a minimum output from a router quote with a bounded slippage tolerance, and cap the per-trigger swap size.

SMC AUDITS

SMC-L-02 LOW Open

Max-wallet limit enables dust griefing

CATEGORY

Denial of service / griefing

LOCATION

GIANT.sol : _transfer (max-hold check)

DESCRIPTION

The max-wallet check requires `balanceOf(to) + amount <= max_Hold`, keyed on the recipient's current balance with no per-sender gating. An attacker can push a victim's balance close to `max_Hold` by sending tokens, after which the victim's further buys or incoming transfers revert.

IMPACT

Temporary and self-costly. The attacker must give away up to about 0.5% of supply, and the victim can still sell to recover. It only bites when the owner has tightened `max_Hold` toward the floor.

PROOF OF CONCEPT

The check reads the recipient balance; a gifted balance near the cap makes the victim's next incoming transfer revert with the max-wallet error.

RECOMMENDATION

Exempt incoming transfers from the holder cap, or apply the cap only to buys from a pair, so a third party cannot push a victim over the limit.

SMC-I-01 INFO Open

Broken input clamp in Processing__Process_Now

CATEGORY

Code quality

LOCATION

GIANT.sol : Processing__Process_Now

DESCRIPTION

The 100% clamp uses `==` (a comparison) instead of `=` (an assignment): `if (Percent_of_Tokens_to_Process > 100){Percent_of_Tokens_to_Process == 100;}`. The clamp does nothing. A value above 100 then makes the processed amount exceed the contract balance and the owner's own call reverts. Owner-only, no fund impact.

RECOMMENDATION

Replace `==` with `=`.

SMC AUDITS

SMC-I-02 INFO Open

decimals() returns uint256 instead of uint8

CATEGORY

Standards / interoperability

LOCATION

GIANT.sol : decimals

DESCRIPTION

`decimals()` returns `uint256`, while the ERC-20 / BEP-20 standard specifies `uint8`. An integrator that decodes a `uint8` reads the value correctly from the 32-byte word in practice, so impact is low, but the deviation can surprise strict consumers.

RECOMMENDATION

Return `uint8`.

SMC-I-03 INFO Open

SwapAndLiquify event reports the wrong swapped amount

CATEGORY

Code quality / off-chain data

LOCATION

GIANT.sol : swapAndLiquify

DESCRIPTION

The event emits `LP_Tokens` in the `tokensSwapped` field, but the amount actually swapped to BNB is `Swap_Tokens = Tokens - LP_Tokens`. Off-chain readers of the event get an incorrect swapped amount.

RECOMMENDATION

Emit `Swap_Tokens` as the `tokensSwapped` argument.

SMC AUDITS

SMC-I-04 INFO Open

Dev fee removed even if the 3 BNB payment fails to arrive

CATEGORY

Code quality

LOCATION

GIANT.sol : Maintenance__Remove_Contract_Fee

DESCRIPTION

The function requires `msg.value == 3 BNB`, but it ignores the result of the transfer to `feeCollector` while still setting the dev fee to zero. If the fixed `feeCollector` rejected the BNB, the dev fee is removed without the payment landing. `feeCollector` is a constant, so the owner cannot force the rejection; impact is low.

RECOMMENDATION

Require the send result: `require(send_BNB(feeCollector, msg.value), "fee transfer failed");`.

SMC-I-05 INFO Open

Ownership transfer drops the new owner's exemptions

CATEGORY

Code quality

LOCATION

GIANT.sol : Maintenance__Ownership_Transfer

DESCRIPTION

The function clears the old owner's fee and limit exemptions but grants none to the new owner. The new owner then pays fees and hits limits until a later call restores them. Self-fixable by the new owner, no user impact.

RECOMMENDATION

Grant `_isLimitExempt`, `_isExcludedFromFee`, and `_isWhiteListed` to the new owner on transfer.

09

09 Remediation status

After the project addresses the findings, SMC Audits re-audits the changes and records the outcome here.

ID	SEVERITY	STATUS	VERIFIED FIX
SMC-M-01	MEDIUM	Open	—
SMC-M-02	MEDIUM	Open	—
SMC-L-01	LOW	Open	—
SMC-L-02	LOW	Open	—
SMC-I-01	INFO	Open	—
SMC-I-02	INFO	Open	—
SMC-I-03	INFO	Open	—
SMC-I-04	INFO	Open	—
SMC-I-05	INFO	Open	—

10

10 Conclusion

GIANT (GTAN) is a constrained fee-token template with no unprivileged path to theft, drain, or freeze.

The main risks are owner-dependent: a zero-fee state that permanently bricks all sells, and standard owner centralization over fees and auto-liquidity. Applying the one-line guard in SMC-M-01 and reducing or locking owner powers would materially improve holder safety. The principal residual risk is the un-renounced owner key.

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

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