



SMART CONTRACT SECURITY RE-AUDIT

Auren Protocol (AU)

BNB Smart Chain

Final Re-Audit Report - Version 2.0

Assessment date 13 September 2026

Contract

0x14D72A655bB00D6cDA641b797EC8B4d060C472EB

Manual Review • Foundry Regression PoCs • Fuzz / Property Testing

Deployment Bytecode Verification • Live BSC State Review

Auren Protocol - Client Re-Audit Summary

Assessment at a Glance

This re-audit verifies the patched Auren Protocol deployment against the findings from the initial Audit Rate Tech assessment and evaluates newly introduced or remaining security risks.

Project	Auren Protocol (AU)
Network	BNB Smart Chain (BSC)
Re-audited contract	0x14D72A655bB00D6cDA641b797EC8B4d060C472EB
Initial audited contract	0x6951822a267f4ce29af8eb0db46843f585aeb352
Assessment status	Completed - remediation verified with remaining findings
Current unresolved	0 Critical • 0 High • 1 Medium • 4 Low • 1 Informational
Regression / targeted tests	15 passed • 0 failed
Fuzz / property executions	4,000 passed

Remediation outcome

- ART-01, ART-02, ART-04, ART-05 and ART-06 were verified as Resolved in the new implementation.
- ART-03 remains Open: automatic token-to-BNB fee conversion still uses amountOutMin = 0.
- ART-07 is materially reduced: the previous blacklist, mutable fee, maxTx, arbitrary pair and asset-recovery controls were removed. A limited launch/whitelist trust assumption remains.
- Four new Low-severity issues were identified during the re-audit: ART-08 through ART-11.

Client conclusion

The new deployment is a meaningful security improvement over the initially audited version. No Critical or High-severity vulnerability was identified in the reviewed scope. However, the contract should not be described as fully remediated because one Medium and four Low findings remain open.

Re-Audit Dashboard

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	0	1	4	1

15	4,000	MATCH
Targeted / regression tests	Fuzz / property executions	Normalized deployed runtime

REMEDIATION STATUS

ID	Original	Current	Status
ART-01	Medium	-	Resolved
ART-02	Medium	-	Resolved
ART-03	Medium	Medium	Open
ART-04	Low	-	Resolved
ART-05	Low	-	Resolved
ART-06	Low	-	Resolved
ART-07	Informational	Informational	Partially Resolved
ART-08	-	Low	New / Open
ART-09	-	Low	New / Open
ART-10	-	Low	New / Open
ART-11	-	Low	New / Open

Interpretation. A passing vulnerability test means the expected security condition or vulnerable behavior was successfully reproduced. It does not mean that the contract is free of the tested issue.

1. Executive Summary

Audit Rate Tech performed a remediation re-audit of the new Auren Protocol token deployment on BNB Smart Chain. The review compared the patched source against the initial audit findings, reproduced security-critical behaviors with Foundry tests, exercised fee and balance invariants with fuzz/property testing, and verified the deployed runtime against a local Solidity 0.8.14 build.

The re-audit confirms that five implementation findings from the initial report were resolved. The prior transfer truncation and transferFrom accounting mismatch are no longer present; the mutable fund-address and claimToken surfaces were removed; and native fund payout no longer relies on Solidity transfer().

ART-03 remains unresolved because the internal PancakeSwap fee conversion continues to set amountOutMin to zero. This permits execution without an on-chain minimum-output bound and remains the highest-severity open item in this re-audit.

The redesign also substantially reduces privileged owner controls. Nevertheless, the re-audit identified four Low-severity operational and standards issues involving pre-launch ownership renunciation, zero-address transfers, ownership/whitelist role migration, and unrecoverable native BNB.

Recommended release posture

The deployment may be described as re-audited with verified remediations and remaining findings. It should not be represented as vulnerability-free, fully remediated, or formally verified.

2. Engagement & Scope

In scope

- AurenProtocol.sol and its included ERC20, router, factory and ownership interfaces/logic.
- Regression verification of ART-01 through ART-07 from the initial audit.
- ERC20 transfer, transferFrom, allowance, zero-address and balance-conservation behavior.
- Buy/sell fee accounting and automatic token-to-BNB fee conversion.
- Ownership lifecycle, launch activation and fee-whitelist behavior.
- PancakeSwap V2 router/pair integration assumptions.
- Selected deployed BSC state and runtime bytecode integrity.

Out of scope

- Website/frontend, private-key custody, off-chain infrastructure and operational procedures.
- PancakeSwap implementation internals beyond the contract's interaction assumptions.
- Market performance, liquidity depth, token price, investment returns or future trading conditions.
- Future source changes, redeployments, role changes or external dependency changes after the snapshot.
- Formal mathematical verification.

Immutable re-audit target

Artifact	Value
Network	BNB Smart Chain
Contract	0x14D72A655bB00D6cDA641b797EC8B4d060C472EB
Reviewed source	AurenProtocol.sol
Solidity	0.8.14
Optimizer	Disabled
Source SHA-256	3fbca209150f9b354f7a9625e3b5268df7ffaf9ef05b45789d14b0dc471629e3
Runtime Keccak-256	0x1ccb678acad5390a5cc8e10ff1ebd2062c2440eb2135680c985bce75c5136d8c

3. Methodology & Verification

Stage	Status
Source acquisition / hashing	Completed
Exact compiler reproduction	Completed
Manual source review	Completed
Foundry regression testing	Completed
Targeted edge-case testing	Completed
Fuzz / property testing	Completed
Runtime bytecode normalization / comparison	Completed - MATCH
Live BSC state checks	Completed
Formal verification	Not performed

Manual review

Business logic, ERC20 semantics, fee arithmetic, external calls, launch controls, ownership transitions, whitelist persistence and native-asset handling were reviewed manually.

Regression and targeted PoCs

Fifteen targeted Foundry tests were executed successfully. Tests covered the initial transfer/accounting findings, zero-slippage swap configuration, swapback sizing, pre-launch restrictions, ownership renunciation, zero-address behavior, whitelist migration and native BNB retention.

Fuzz / property testing

Four properties were executed for 1,000 generated cases each: buy-fee accounting, sell-fee accounting, buy balance conservation and sell balance conservation. Total: 4,000 passing executions.

Deployment integrity

The on-chain runtime and locally compiled runtime were compared after masking immutable references and removing Solidity compiler metadata. The normalized executable runtime matched byte-for-byte. This is deployment integrity verification, not formal verification.

4. Architecture & Attack Surface

The patched design is materially simpler than the initially audited contract. The primary value and trust flow is:

```
USER → AUREN TOKEN → PANCAKESWAP V2 PAIR
      ↓ fee tokens
AUTOMATIC SWAPBACK → BNB → IMMUTABLE FUND ADDRESS
```

Security boundaries

- Users depend on standard ERC20 transfer and allowance behavior implemented by AUREN.
- AMM-related transfers remain gated by startTradeBlock for non-whitelisted addresses before public activation.
- Fee conversion depends on the immutable PancakeSwap V2 router and current market execution conditions.
- The fund address is immutable in the deployed implementation.
- The original deployer/owner is fee-whitelisted at construction; ownership transfer does not migrate that whitelist state.
- The contract accepts native BNB through receive(), while only current-swap BNB deltas are distributed.

Removed privileged surfaces compared with the initial version

The re-audited ABI contains no setBlackList, removeFromBlackList, setFee, setSwapPairList, setMaxTxAmount, setFundAddress or claimToken functions. This significantly narrows the post-deployment administrative attack surface.

5. Findings Summary

ID	Severity	Status	Finding
ART-01	Medium	Resolved	Silent reduction of full-balance transfers
ART-02	Medium	Resolved	transferFrom allowance/value mismatch
ART-03	Medium	Open	Automatic fee swaps execute without a minimum output
ART-04	Low	Resolved	Fee receiver failure via Solidity transfer()
ART-05	Low	Resolved	Previous fund addresses retain whitelist privileges
ART-06	Low	Resolved	claimToken() ignores ERC20 transfer result
ART-07	Info	Partially Resolved	Privileged control / remaining launch trust assumptions
ART-08	Low	Open / New	Renouncing ownership before launch can permanently lock trading
ART-09	Low	Open / New	Transfers to address(0) are allowed without reducing supply
ART-10	Low	Open / New	Ownership transfer does not migrate fee-whitelist privileges
ART-11	Low	Open / New	Direct native BNB can become permanently unrecoverable

Current unresolved severity: 0 Critical • 0 High • 1 Medium • 4 Low • 1 Informational.

Resolved findings remain documented because this is a remediation re-audit. Their inclusion demonstrates what changed and what evidence was used to verify the fix.

6. Verified Remediations

ART-01 - Full-Balance Transfer Truncation

Status: **Resolved** Original severity: Medium

The previous implementation silently reduced certain requested transfers to 99.99% of balance. The patched `_transfer()` no longer applies the `maxSellAmount` truncation. The regression test confirmed that a full-balance transfer succeeds with the complete requested value.

ART-02 - `transferFrom` Accounting Mismatch

Status: **Resolved** Original severity: Medium

Because transfer amounts are no longer silently reduced, `transferFrom()` allowance consumption and delivered value remain consistent. The dedicated regression test passed.

ART-04 - Fee Receiver Failure via `transfer()`

Status: **Resolved** Original severity: Low

The old 2,300-gas Solidity `transfer()` payout was replaced with a low-level call and success validation. The deployed immutable fund address was also observed to be an EOA at the assessment snapshot.

ART-05 - Historical Fund Whitelist Persistence

Status: **Resolved** Original severity: Low

`setFundAddress()` was removed and `fundAddress` is immutable, eliminating fund rotation and the prior accumulation of historical fee-whitelisted fund addresses.

ART-06 - `claimToken` Return Value

Status: **Resolved** Original severity: Low

`claimToken()` was removed from the patched contract, eliminating the affected recovery path.

MEDIUM • OPEN

ART-03 - Automatic Fee Swaps Execute Without a Minimum Output

Category	DEX / MEV / Slippage
Affected component	AUREN.swapTokenForFund()
Re-audit evidence	test_ART03_ZeroSlippageProtection() - PASS
Status	Open - reproduced in patched implementation

Description

The patched contract continues to pass zero as amountOutMin when swapping accumulated AU fee tokens for native BNB through the PancakeSwap-compatible router.

Affected logic

```
_swapRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(  
    tokenAmount - lpAmount,  
    0,  
    path,  
    address(this),  
    block.timestamp  
);
```

Impact

The swap can accept an arbitrarily poor positive output. During volatile or manipulated market conditions, fee conversion may execute at a materially unfavorable price and may be exposed to adverse MEV ordering.

Recommendation

Introduce a bounded slippage policy based on a defensible quote/oracle mechanism, or redesign fee conversion so that execution is not forced without a meaningful minimum output.

FO • PARTIALLY RESOLV

ART-07 - Privileged Control and Remaining Launch Trust Assumptions

The initial contract exposed extensive owner controls over blacklist state, fees, max transaction size, pair classification, fund rotation and asset recovery. Those broad mutable controls are absent from the re-audited ABI.

Remaining trust assumptions

- Only the owner can call startTrade(), and startTrade() is a one-shot activation.
- The original deployer/owner is fee-whitelisted in the constructor.
- Fee-whitelisted addresses can interact with the registered pair before public trading activation.
- Ownership transfer changes _owner but does not automatically migrate the deployer's whitelist state; this is separately reported as ART-10.

Impact

The centralization surface is substantially reduced compared with the initial deployment, but launch timing and pre-launch whitelist privileges remain trust assumptions that should be disclosed to users.

Recommendation

Document launch authority and whitelist behavior. If ownership is expected to represent the active privileged operator, align whitelist lifecycle with ownership lifecycle.

LOW • NEW / OPEN

ART-08 - Ownership Can Be Renounced Before Trading Activation

Category	Availability / Ownership Lifecycle
Affected components	Ownable.renounceOwnership(), AUREN.startTrade()
Evidence	test_RenounceBeforeStartPermanentlyDisablesStartTrade() - PASS

Description

renounceOwnership() can be called while startTradeBlock is still zero. After ownership is renounced, the owner becomes address(0), while startTrade() remains protected by onlyOwner.

Impact

If the owner renounces ownership before public trading is activated, no account can subsequently call startTrade(). Non-whitelisted pair interaction remains permanently blocked. This is an owner-triggered operational failure rather than an external attacker exploit.

Recommendation

Prevent ownership renunciation until trading has been activated, for example by requiring startTradeBlock != 0 before renounceOwnership(), or enforce an operational launch sequence that cannot leave the contract permanently unstartable.

LOW • NEW / OPEN

ART-09 - Transfers to the Zero Address Are Allowed Without Reducing Supply

Category	ERC20 / Standards / Asset Loss
Affected component	AUREN._transfer()
Evidence	test_TransferToZeroAddressSucceeds() - PASS

Description

_transfer() does not reject address(0) as the recipient. A user can successfully transfer AU to the zero address; the zero-address balance increases while totalSupply remains unchanged.

Impact

Users or integrations can irreversibly lock tokens through a successful transfer that resembles a burn but does not reduce total supply. This deviates from the common OpenZeppelin ERC20 behavior and can create accounting ambiguity.

Recommendation

Reject zero-address recipients in _transfer(), e.g. require(to != address(0), "transfer to zero"). If burning is intended, implement an explicit burn mechanism that also reduces totalSupply.

Additional note

approve(address(0), amount) also succeeds. This was reproduced but is not assigned a separate finding because it does not create an independently exploitable path in the reviewed design.

LOW • NEW / OPEN

ART-10 - Ownership Transfer Does Not Migrate Fee-Whitelist Privileges

Category	Access Control / Role Consistency
Affected components	Ownable.transferOwnership(), _feeWhiteList
Evidence	4 dedicated ownership/whitelist tests - PASS

Description

The constructor permanently fee-whitelists msg.sender. transferOwnership() updates only _owner; it does not remove the old owner's whitelist entry or grant the new owner a corresponding whitelist entry.

Observed behavior

- Old owner remains fee-whitelisted after ownership transfer.
- New owner is not automatically fee-whitelisted.
- Old owner retains pre-launch pair interaction privilege.
- New owner cannot use the pair pre-launch unless independently whitelisted.

Impact

Ownership and fee/launch privileges can diverge. Operational teams may assume that transferring ownership transfers the active privileged role, while the previous owner retains persistent fee exemption and pre-launch pair access.

Recommendation

Explicitly migrate owner-linked whitelist privileges during ownership transfer, or document that deployer whitelist status is independent and permanent by design.

LOW • NEW / OPEN

ART-11 - Direct Native BNB Can Become Permanently Unrecoverable

Category	Native Asset Handling / Operational Risk
Affected components	receive(), swapTokenForFund()
Evidence	Direct-BNB and pre-existing-BNB swapback tests - PASS

Description

The token contract accepts native BNB through receive(), but exposes no native-asset rescue or withdrawal function. swapTokenForFund() records BnbBalanceBefore and distributes only the increase generated by the current swap.

Affected behavior

fundAmount is derived from BnbBalanceNow - BnbBalanceBefore. BNB already present before swapback is therefore excluded from distribution and remains on the token contract.

Impact

BNB accidentally or deliberately sent directly to the token contract can become permanently inaccessible. At the assessment snapshot, the deployed token contract held 0 BNB, so no current loss was observed.

Recommendation

Add a carefully restricted native-asset recovery mechanism, or redesign native-BNB acceptance so unintended direct transfers cannot create unrecoverable balances. Any recovery mechanism should be documented as a privileged capability.

7. Test Evidence

Test area	Result
ART-01 full-balance regression	PASS
ART-02 transferFrom accounting regression	PASS
ART-03 zero minimum output	PASS - issue reproduced
Swapback sizing / fund distribution	PASS
Pre-launch normal-user restriction	PASS
Pre-launch whitelisted owner behavior	PASS
Renounce-before-start ownership edge case	PASS - issue reproduced
Zero-address transfer	PASS - issue reproduced
Zero-address approval	PASS - note reproduced
Ownership/whitelist migration suite	4 / 4 PASS
Direct native BNB retention	PASS - issue reproduced
Pre-existing BNB during swapback	PASS - issue reproduced

Targeted / regression total: 15 PASS • 0 FAIL

Fuzz / property total: 4,000 PASS

Property	Runs
Buy fee accounting	1,000 PASS
Sell fee accounting	1,000 PASS
Buy balance conservation	1,000 PASS
Sell balance conservation	1,000 PASS

Fork-dependent regression/pre-launch tests were successfully executed against a historical BSC state during the assessment. Later archive-node limitations do not represent test failures.

8. On-chain Snapshot

Parameter	Observed value
Owner	0xF918D4e107dE957bb890E420f523B862c5c01Df1
Fund address	0xF918D4e107dE957bb890E420f523B862c5c01Df1
Fund address code	0x - EOA at snapshot
Router	0x10ED43C718714eb63d5aA57B78B54704E256024E
AU/WBNB pair	0x8B192fa61f21E164b1c26E9A75B0876e6188f001
Buy fund fee	300 bps (3.00%)
Sell fund fee	800 bps (8.00%)
Buy / sell LP fee	0 / 0 bps
totalSupply	1e27 base units
startTradeBlock	0
Pair reserves	0 AU / 0 WBNB
Pair LP total supply	0
Token contract AU balance	0
Token contract BNB balance	0

At the snapshot, the pair existed but no liquidity had been added and public trading had not been activated. The token contract held neither accumulated AU fees nor native BNB.

These observations are point-in-time values. Future launch actions, liquidity provisioning, ownership changes or market activity can change the deployed state without changing the bytecode.

9. Static / Lint Triage

Foundry compiler/lint output was manually reviewed. Scanner warnings were not automatically promoted to findings.

Observation	Assessment
Reentrancy warnings around router/fund calls	Manually triaged; no new deployed finding. lockTheSwap sets inSwap before the swap body; router
Default-initialized locals	False positive / language semantics; bool and uint locals initialize to false / 0.
Missing ownership events	False positive; OwnershipTransferred is emitted before owner changes.
Unused addLiquidityETH return values	No demonstrated security impact; current LP fee constants are zero.
Constructor zero-address checks	Hardening note. Router, fund and receiver parameters lack explicit zero checks, but deployed valu
Unlimited router allowance	Reviewed as standard swapback pattern; allowance is from the token contract's own balance to th
Naming / style / custom error suggestions	Code-quality only.

Constructor hardening note. RouterAddress, FundAddress and ReceiveAddress are not explicitly checked against address(0). This is not assigned a separate finding for the reviewed deployment because the deployed immutable/configured values are valid.

10. Initial Audit → Re-Audit Matrix

ID	Initial issue	Re-audit evidence	Final status
ART-01	Silent transfer truncation	Full-balance regression PASS	Resolved
ART-02	Allowance/value mismatch	transferFrom regression PASS	Resolved
ART-03	amountOutMin = 0	Mock-router capture PASS	Open / Medium
ART-04	transfer() payout DoS	call() used; fund is EOA	Resolved
ART-05	Mutable fund whitelist persistence	Fund immutable; setter removed	Resolved
ART-06	Unchecked claimToken transfer	claimToken removed	Resolved
ART-07	Broad owner controls	Mutable admin functions removed; launch privilege partially resolved	Partially Resolved / Info
ART-08	New	Ownership edge-case PoC PASS	Open / Low
ART-09	New	Zero-address transfer PoC PASS	Open / Low
ART-10	New	4 ownership/whitelist tests PASS	Open / Low
ART-11	New	2 native-BNB tests PASS	Open / Low

Priority recommendation

P1: remediate ART-03 before relying on automatic fee conversion under meaningful market liquidity.
P2: address ART-08 through ART-11 as part of launch hardening and standards/operational cleanup.
ART-07 should remain documented as a trust-model disclosure.

Appendix A - Verification Record

Artifact	Value
Re-audited contract	0x14D72A655bB00D6cDA641b797EC8B4d060C472EB
Initial audited contract	0x6951822a267f4ce29af8eb0db46843f585aeb352
Source SHA-256	3fbca209150f9b354f7a9625e3b5268df7ffaf9ef05b45789d14b0dc471629e3
On-chain runtime Keccak-256	0x1ccb678acad5390a5cc8e10ff1ebd2062c2440eb2135680c985bce75c5136d8c
Compiler	Solidity 0.8.14
Optimizer	Disabled
Normalized executable length	9,035 bytes
Runtime comparison	MATCH after masking immutables and stripping compiler metadata
Targeted / regression tests	15 PASS / 0 FAIL
Fuzz / property executions	4,000 PASS
Historical successful fork block	121613796
Formal verification	Not performed

Runtime verification interpretation

The deployed runtime was reproduced from the reviewed source using Solidity 0.8.14 with the optimizer disabled. Immutable references and compiler metadata were normalized before comparing executable runtime bytes. The resulting executable runtime matched. This establishes strong deployment/source consistency for the reviewed artifact, but it is not a mathematical proof of contract correctness.

Appendix B - Disclaimer

This report is a point-in-time security assessment of the reviewed smart-contract source and selected deployed state. A security audit cannot guarantee the absence of all defects, economic attacks, operational failures, private-key compromise, third-party failures or future vulnerabilities.

The assessment does not constitute financial, investment, legal or regulatory advice. Changes to source code, deployment configuration, privileged roles, liquidity conditions, external dependencies or operational practices may invalidate parts of this report.

The term Final Re-Audit Report means this remediation assessment document is finalized for the reviewed snapshot. It does not mean all findings are resolved. Finding status is stated individually throughout this report.

Audit Rate Tech did not perform formal mathematical verification and did not execute live mainnet trades using real user funds as part of this assessment.

Audit Rate Tech

Smart Contract Security Assessment
Auren Protocol (AU)
Final Re-Audit Report v2.0 - 13 September 2026

Website: auditrate.tech
Email: support@auditrate.tech
Smart Contract Security Since 2021