NC-TOKEN AUDIT REPORT

DATE	SEVERITY	LOCATION	TITLE	SCREENSHOT IMAGE	DESCRIPTION	SOLUTION
10-1-26	CRITICAL	spl-project, mint-tokens	Blacklist user can mint tokens	screenshot	Governance can mint tokens to blacklisted addresses,	Add blacklist check for the recipient before minting.
10-1-26	CRITICAL	spl-project, burn-tokens	Unrestricted Token Burning	screenshot	The burn_tokens function lets tokens be burned using any external signer rather than through the program's governance PDA,Although the function verifies that the caller (governance) matches state. authority.	strict authorization by ensuring that token burning can only be executed by the governance PDA
10-1-26	CRITICAL	spl-project, transfer-tokens	not ready for production	screenshot	transfer_tokens function, in its current form, is not ready for production because it skips several critical security and business logic checks. Blacklist, restricted, and liquidity pool validations are not enforced, the sell limit mechanism is incomplete and does not check actual balance	
10-1-26	CRITICAL	governance,approve_transaction	Missing Cooldown Enforcement	screenshot	Auto-execution check in approve_transaction uses Clock::get() to compare transaction.execute_after, but this is misleading since the real execution happens in execute_transaction	Remove cooldown check from approve_transaction. Only enforce cooldown in execute_transaction.
10-1-26	CRITICAL	govenance,approve_transaction	Auto-Execution (Approve)	screenshot	approve_transaction marks transaction.status = Executed when approvals meet the condition without performing actual execution.	Execution should only occur via execute_transaction.
	CRITICAL	governanace,approve_transaction	Misleading Transaction Status		approve_transaction may mark a transaction as Executed even though execute_transaction hasn't been called.	Only update status in execute_transaction
10-1-26	CRITICAL	Governance,execute_ transaction	required account missing		Certain transaction types (Blacklist, NoSellLimit, Restrict, Pair) are logged only; missing required PDAs.May be gonna implement in next stage	
10-1-26	CRITICAL	presale - buy and withdraw_to_treasury	Bypassing Anchor Type Safety	screenshot	uses UncheckedAccount for buyer/payment token accounts and manually reads account data to extract mint and owner. This bypasses Anchor's type checks.No validation that account is a token account	Use proper Account<'info, TokenAccount> for all token accounts.
10-1-26	CRITICAL	PRESale - buy and withdraw_to_treasury	Unsafe Direct Memory Reads	screenshot	If the account is malformed or not a token account, this could read garbage data.	use Anchor Account<'info, TokenAccount> to safely deserialize and validate accounts.
10-1-26	CRITICAL	spl-project-set_governance	No multi-sig implementation		Only the current authority is verified;Whoever is the current authority can instantly set new_authority to any address, including a wrong or compromised address.	Use multi-sig or require multiple approvals for governance changes
10-1-26	MEDIUM	presale - set_treasury_address	Treasury Address Validation Missing	screenshot	no check if it is valid address	add validation
10-1-26	LOW	governance - queue_set_cooldown_period	No Maximum Cooldown Period	screenshot	NO maximum limit, could set any limit	Add maximum limit
10-1-26	LOW	presale,buy	Hardcoded numbers	screenshot	numbers should be a named constant explaining the offset calculation.	
10-1-26	LOW		Missing Events		Token mints/burns,treasury withdrawls,paused	
10-1-26	LOW	presale-buy	No Presale Cap or Limits		There's no maximum presale cap, per-user limits,	
10-1-26	INFORMATIONAL		Emergency pause implementation		Emergency pause flag is only checked in transfer_tokens.Other functions like mint_tokens are not paused.	Apply emergency pause check in all mutable functions that affect token balances.