Light Audit Report
ERC20 – PumpDaddy Token

Auditor: Febri Nirwana
Date: July 2025
Network: Ethereum
Contract Address:
0x29FfAEfB147C57dB2B787766eD9bb5b0fDbA3ff8

1. Disclaimer

This is an independent light audit conducted based on publicly available source code. This report does not guarantee the absence of bugs or vulnerabilities. Always do your own due diligence.

2. Scope of Review

Contract Reviewed: `PumpDaddy.sol`
Code Source: Etherscan (verified source)
Commit / Block Date: 2025-07-15
Review Focus: Logic correctness, security patterns, custom implementations

3. Summary of Findings

No	Severity	Issue
1	Medium	Public setter for private variable `var_SeHBuB()`
2	Medium	Custom ERC20 logic without standard libraries
3	Low	Unused function `add_NnuxW()` writing to storage
4	Neutral	No ownership or control functions implemented

4. Detailed Findings

a. Finding 1 – Suspicious update_var_SeHBuB() Public Setter

Severity: Medium
Location: update_var_SeHBuB()
Description:
This function allows anyone to modify a private storage variable (var_SeHBuB). Although the variable has no effect on token logic now, it could be used later as a hidden trigger in a connected dApp or off-chain logic.
Why it's dangerous:
The variable name is obfuscated. This suggests potential use as a backdoor, hidden condition, or honeypot flag.
Recommendation:
Remove the setter if unused, or restrict it with access control like onlyOwner.

b. Finding 3 – Custom ERC20 Without Standard Libraries

Severity: Medium
Location: Entire contract
Description:
The contract manually implements ERC20 logic such as transfer, approve, and transferFrom without using OpenZeppelin or any security-hardened library.
Why it's risky:
This increases potential for logic bugs, compatibility issues, and missed best practices (e.g., no SafeMath, no standard mint/burn, no Ownable).

Recommendation:
Use OpenZeppelin ERC20 implementation to ensure best practice, gas optimization, and compatibility.

c. **Finding 2 – Dummy Function add_NnuxW() Writing to Private Storage**
Severity: Low
Location: add_NnuxW()
Description:
The function allows anyone to write to a private variable var_mDdktX without any usage in the contract. The value is stored but never read, affecting gas and storage.
Why it's suspicious:
It could mislead users, simulate activity, or be misused in frontend UI logic.
Recommendation:
Remove the function entirely unless it serves a future purpose. Avoid unexplained write operations to private variables.

d. **Observation – Minimal Token Contract With No Owner**
Severity: Neutral
Description:
There is no ownership mechanism, no mint/burn logic, and no special privileges coded. The token appears to be fixed-supply with no future control expected.
Recommendation:
None. This is acceptable if intentional.

5. Recommendations (Global Summary)

a. Remove or restrict `update_var_SeHBuB()`
b. Replace Manual ERC20 Implementation with OpenZeppelin
c. Delete `add_NnuxW()` function
d. Clarify Token Purpose and Roadmap

6. Final Notes

The The PumpDaddy contract appears to be a simple, fixed-supply ERC20 token with no owner privileges or advanced features. However, the presence of obfuscated variable names and publicly writable storage creates unnecessary risk and confusion.

There is no malicious behavior confirmed, but the design choices and unused functions indicate a lack of code hygiene and purpose clarity.

We recommend **not using this contract in production** without significant refactoring and clear documentation on its intended use.