VulnerableToken Smart Contract Audit

Unsafe approve & Owner backdoor

CRITICAL SEVERITY

Score: 9/10

Prepared by: Abdelhamid Charir (CH Abdelhamid)
Lead Auditor: CHARIR Abdelhamid
Date: October 22, 2025

Confidential Security Assessment Report

Contents

1 Executive Summary **2**

2 Affected Files **2**

3 Proof of Concept / Reproduction **2**
3.1 Test Execution 2

4 Automated Analysis **2**
4.1 Slither Results 2

5 Impact Assessment **3**

6 Remediation **3**

7 Regression Tests **3**

8 Severity Assessment **4**

1 Executive Summary

The contract implements ERC-20 functionality but contains two critical issues:

1. `approve` uses the classic set pattern that is susceptible to allowance-race issues
2. `adminTransferFrom` is an owner-only backdoor that allows the owner to transfer tokens from any account without consent

Critical Impact

Impact: Full fund compromise by owner key compromise or malicious owner.
Severity: Critical

2 Affected Files

- `src/VulnerableToken.sol` (lines showing `approve`, `transferFrom`, and `adminTransferFrom`)

3 Proof of Concept / Reproduction

In `test/Exploit_AdminBackdoor.t.sol`, especially in the `testAdminBackdoorExploit()` function (see: https://vscode.dev/github/CHARIRAbdelhamid/vulner-erc-20/blob/master/test/Exploit_AdminBackdoor.t.sol#L22), we call the `adminTransferFrom` function to move 500 tokens from alice to bob.

3.1 Test Execution

After running the command:

```
forge test --match-test testAdminBackdoorExploit -vvv
```

You should see the following output:

```
[] Compiling...
No files changed, compilation skipped

Ran 1 test for test/Exploit_AdminBackdoor.t.sol:AdminBackdoorExploitTest
[PASS] testAdminBackdoorExploit() (gas: 47304)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 4.09ms (488.60
    µs CPU time)

Ran 1 test suite in 46.84ms (4.09ms CPU time): 1 tests passed, 0 failed, 0
    skipped (1 total tests)
```

4 Automated Analysis

4.1 Slither Results

ID	Title	Severity	Description	Recommendation
S1	**Use of Solidity version ^0.8.19 with known compiler issues**	Medium	Slither detected that the contract uses Solidity version `^0.8.19`, which contains known issues such as `VerbatimInvalidDeduplication`, `FullInlinerNonExpressionSplitArgumentEvaluation` and `MissingSideEffectsOnSelectorAccess`.	Update to a safer compiler version (e.g., `^0.8.23` or latest stable).
S2	**Non-conforming parameter naming**	Informational	The parameter `_owner` in `allowance(address,address)` does not follow mixedCase convention.	Rename to `owner_` or `ownerAddress` to conform with Solidity naming standards.
S3	**Variables could be declared constant**	Low	The variables `decimals`, `name`, and `symbol` are never modified.	Declare them as `constant` to save gas and improve readability.
S4	**Variable could be declared immutable**	Low	The variable `owner` is assigned only once in the constructor.	Declare it as `immutable` for gas optimization and stronger guarantees.

Table 1: Slither Static Analysis Findings

5 Impact Assessment

Critical Risk

Funds at risk: Any user token balance can be moved by the owner; effectively all funds.

Difficulty: Low (requires only owner private key or malicious owner behavior)

6 Remediation

1. **Remove** `adminTransferFrom`. If a recovery mechanism is required, implement it behind a secure multisig and timelock and emit detailed events for audit.
2. **Adopt safe allowance patterns** — either require zero-first updates, or implement `increaseAllowance`/`decreaseAllowance` helpers (see OpenZeppelin ERC20).
3. **Use role-based access control** (OpenZeppelin `AccessControl` or multisig) for minting and privileged operations.

7 Regression Tests

Add tests verifying `adminTransferFrom` is removed and `approve` behavior includes zero-first requirement or uses increase/decrease helpers.

8 Severity Assessment

Level: Critical
Score: 9/10

End of Security Audit Report