// Security Assessment

07.24.2025 - 07.25.2025

# EVM Token

## *Paxos*

HALBORN

# EVM Token - Paxos

Prepared by: HALBORN

Last Updated 08/01/2025

Date of Engagement: July 24th, 2025 - July 25th, 2025

## Summary

100% OF ALL REPORTED FINDINGS HAVE BEEN ADDRESSED

| ALL FINDINGS | CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 |

## TABLE OF CONTENTS

## 1. Summary

Paxos Global contracted Halborn to conduct a targeted security review of its Ethereum-based PaxosToken contracts, focusing exclusively on the code changes introduced in commit `2802943b53e3d698c3d29cf6f913c968b64594fa` . The engagement took place from June 23 to June 24, 2025.

## 2. Assessment Summary

A senior Halborn blockchain security engineer was dedicated full-time to this review. The objectives were to:

- Verify the correctness of the new domain separator initialization logic during deployment, upgrades, and runtime.
- Identify vulnerabilities that could lead to fund loss, privilege escalation, or denial-of-service attacks.
- Ensure that the newly added tests do not introduce hidden risks or compromise existing invariants.

## 3. Test Approach & Methodology

- **Change-focused diff analysis:** Reviewed all added, removed, or modified lines resulting from the commit.
- **Manual line-by-line inspection:** Traced control flow, storage modifications, and role verification in PaxosTokenV2.sol.
- **Invariant verification:** Confirmed that balances, total supply, and role assignments remain consistent.
- **Static analysis:** Employed Slither and custom scripts to detect common Solidity pitfalls.

## 4. RISK METHODOLOGY

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the LIKELIHOOD of a security incident and the IMPACT should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

### RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

### RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.


- **10** - CRITICAL
- **9 - 8** - HIGH
- **7 - 6** - MEDIUM
- **5 - 4** - LOW
- **3 - 1** - VERY LOW AND INFORMATIONAL

Our penetration tests use the industry standard Common Vulnerability Scoring System (CVSS) to calculate the severity of our findings.

## 5. SCOPE

Out-of-Scope: New features/implementations after the remediation commit IDs.

## 6. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

| CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |

| SECURITY ANALYSIS | RISK LEVEL | REMEDIATION DATE |
|---|---|---|
| UNRESTRICTED INITIALIZEDOMAINSEPARATOR() | INFORMATIONAL | ACKNOWLEDGED - 08/01/2025 |

# 7. FINDINGS & TECH DETAILS

## 7.1 UNRESTRICTED INITIALIZEDOMAINSEPARATOR()

// INFORMATIONAL

### Description

Commit `2802943b53e3d698c3d29cf6f913c968b64594fa` adds a public function in PaxosTokenV2.sol that recomputes and stores the `DOMAIN_SEPARATOR`:

```
function initializeDomainSeparator() public {
    _initializeDomainSeparator();
}

function _initializeDomainSeparator() private {
    DOMAIN_SEPARATOR = EIP712._makeDomainSeparator(name(), "1");
}
```

Because the function has **no access-control modifier**, any account can call it an unlimited number of times.

- Each call overwrites the same storage slot, burning ~5 000 gas for the caller but leaving contract state unchanged.
- After a future network chainId change, the first arbitrary caller could rewrite the separator and inadvertently invalidate every EIP-2612 / EIP-3009 signature created on the old chain-ID, producing a denial-of-service for off-chain approvals.

Current impact is therefore limited to unnecessary gas consumption and a *potential* DoS scenario following a chain-ID change, no loss of funds or privilege escalation under normal conditions.

### Score

(0.0)

### Recommendation

It is recommended to add an access-control modifier (e.g., `onlyRole(DEFAULT_ADMIN_ROLE)`).

### Remediation Comment

**ACKNOWLEDGED:** The **Paxos team** acknowledged this finding.

---