HACKEN

# Blockchain Protocol Security Analysis Report

**Customer:** Dione Protocol

**Date:** 26/04/2024

We express our gratitude to the Dione Protocol team for the collaborative engagement that enabled the execution of this Security Assessment of the project's implementation.

**Platform:** Dione Protocol

**Language:** Golang

**Timeline:** 26/03/2024 - 26/04/2024

**Methodology:** Blockchain Protocol and Security Analysis Methodology

## Review Scope

| | |
|---|---|
| Repository | https://github.com/DioneProtocol/odysseygo |
| Commit | b44df2531bd9b33cbc8e778f64f1f8cfb5d8c602 |
| Remediation Commit | a89aef3b66c01cd02ce1d32194655386f9747a77 |
| Repository | https://github.com/DioneProtocol/coreth |
| Commit | f9d2ba69b402b761854f09413cb270110e65333c |
| Remediation Commit | 4ed7a414f7f5175c91d300069da5d56195323efc |
| Repository | https://github.com/DioneProtocol/sfxdx__orion-sc |
| Commit | 5755426b9d309993d96b27c6eb4240d36249deb9 |

## Audit Summary

| 10/10 | 9/10 | 10/10 | 10/10 |
|---|---|---|---|
| Security Score | Code quality score | Architecture quality score | Documentation quality score |

# Total 9.5/10

The system users should acknowledge all the risks summed up in the risks section of the report

| 2 | 2 | 0 | 0 |
|---|---|---|---|
| Total Findings | Resolved | Accepted | Mitigated |

### Findings by severity

| | |
|---|---|
| Critical | 0 |
| High | 0 |
| Medium | 1 |
| Low | 1 |

| Vulnerability | Status |
|---|---|
| F-2024-2036 - Update Outdated External Dependencies in Coreth and Odysseygo Repositories | Fixed |
| F-2024-2042 - Potential Exposure of Staking Node Keys | Fixed |

This report may contain confidential information about IT systems and the intellectual property of the Customer, as well as information about potential vulnerabilities and methods of their exploitation.

The report can be disclosed publicly after prior consent by another Party. Any subsequent publication of this report shall be without mandatory consent.

## Document

| | |
|---|---|
| Name | Blockchain Protocol Code Review and Security Analysis Report for Dione Protocol |
| Audited By | Yaroslav Bratashchuk |
| Approved By | Luciano Ciattaglia |
| Website | https://www.dioneprotocol.com/ |
| Changelog | 06/04/2024 - Preliminary Report<br>26/04/2024 - Final Report |

# Table of Contents

## System Overview

Odyssey Chain is a composite of three primary blockchains (Delta, Alpha, and Omega Chains), each fulfilling specific roles in asset creation, trading, and smart contract execution within the ecosystem.

Infi-Nets are modular networks that extend the core architecture, providing scalability and customization for different business requirements.

Parent Network: Acts as a central Infi-Net containing all validators, which includes those from every other Infi-Net, facilitating network-wide governance and coordination.

Delta Chain (D): Supports Ethereum Virtual Machine (EVM) compatible contracts, focusing on Solidity-based dApps, easing developer onboarding and integration.

Alpha Chain (A): Manages the issuance and exchange of Dione coins along with other digital assets, enabling the creation of tokens, NFTs, and stablecoins.

Omega Chain (O): Offers infrastructure for launching and customizing Infi-Nets, allowing for unique blockchain rules and logic definitions by developers.

Customization and Independence: Infi-Nets provide tools for developers to tailor virtual machines, tokenomics, validator requirements, and security settings, ensuring each network can operate as a standalone entity.

Network Efficiency: Validators can be part of multiple Infi-Nets, maintaining performance and compliance with each network's specific rules, thus preventing any single Infi-Net's issues from affecting others.

Private Networks and Efficiency: The Odyssey Chain's architecture supports the creation of private networks with specific validator rules, promoting efficient transaction processing and reduced network congestion.

# Executive Summary

This report presents an in-depth analysis and scoring of the customer's newly developed blockchain protocol project.

Initially, we encountered significant challenges with the previous code, including unnecessary layers, critical bugs, and errors introduced by earlier developers. These issues were so severe that we were unable to continue with their codebase. Consequently, this report is based on a completely new code generated from scratch, reflecting both the innovative approach and the rigorous standards now implemented.

Detailed scoring criteria can be referenced in the corresponding section of the Blockchain Protocol and Security Analysis Methodology.

## Documentation quality

The total Documentation Quality score is **10** out of **10**.

Repositories features clear and comprehensive documentation that effectively details the implemented changes. While the documentation is user-friendly and well-organized, it could be further improved by repairing three broken links and updating one outdated link in the **odysseygo** to ensure complete and accurate resource accessibility.

## Code quality

The total Code Quality score is **9** out of **10**.

At the start of our audit on November 20, 2023, we encountered significant code quality issues within the **odysseygo** and **coreth** repositories. These issues ranged from documentation inconsistencies to compilation errors. Specifically:

- Both repositories exhibited numerous compilation errors. In **coreth**, unused imports and undefined constants were prevalent, while in **odysseygo**, interface implementation errors and syntax mistakes were common.
- Numerous unit and end-to-end tests were failing, indicating underlying issues within the codebase.

In response to our initial feedback, DioneProtocol temporarily halted the audit to rectify these deficiencies, leading to a significantly improved codebase built from a clean fork of the Avalanche repositories, with the latest updates integrated. This updated version demonstrated considerable improvements:

- No serious flaws or issues were found, indicating a significant enhancement in code stability and functionality.
- We successfully ran the testnet, verifying the intended code behavior, and executed end-to-end, unit, and fuzz tests effectively.

To further improve the codebase and ensure ongoing quality and security, we recommend:

- Establish a comprehensive CI pipeline to automate the detection and rectification of build failures, test suite issues, and linting problems.
- Set up continuous fuzzing for the existing fuzz tests to proactively identify and mitigate potential vulnerabilities and logic errors.
- Address all testing-related observations mentioned in this report to ensure comprehensive code quality and reliability.

Overall, we are satisfied with the quality of the repositories, recognising the significant strides made in improving the codebase's integrity and operational efficiency.

## Architecture quality

The total Architecture Quality score is **10** out of **10**.

The Odyssey Chain's design is strong and makes sense. The way it handles money, rewards, and fees is smart and doesn't have any big problems. Its use of validators and delegators, which are common in blockchain, works well and is similar to other successful projects. The method it uses to reach agreement, or consensus, is also well done. The overall design, which includes different types of networks, is well thought out. Any changes made to the original code fit in nicely and are well integrated, showing that the system's design is solid and well-planned.

## Security score

Upon auditing, the code was found to contain **0** critical, **0** high, **1** medium, and **1** low severity issues, leading to a security score of **10** out of **10**.

All identified issues are detailed in the “Findings” section of this report.

## Summary

The comprehensive audit of the customer's blockchain protocol yields an overall score of **9.5**. This score reflects the combined evaluation of documentation, code quality, architecture quality, and security aspects of the project.

# Findings

## Vulnerability Details

### F-2024-2036 - Update Outdated External Dependencies in Coreth and Odysseygo Repositories - Medium

**Description:**

A review of the Coreth and Odysseygo repositories external dependencies has revealed exposure to CVE-2023-44487 and CVE-2023-39325.

The affected dependencies are:
- **golang.org/x/net**
  - Affected versions: < 0.17.0
  - Current version in use: v0.8.0

- **google.golang.org/grpc**
  - Affected versions: < 1.56.3
  - Current version in use: v1.55.0

These outdated dependencies pose a security risk and should be promptly updated to secure versions.

**Assets:**

- Coreth fork review

**Status:** Fixed

#### Classification

**Severity:** Medium

**Impact:** 2/5

**Likelihood:** 2/5

#### Recommendations

**Remediation:**

1. **Update golang.org/x/net**:
   Upgrade the `golang.org/x/net` dependency to at least version 0.17.0 to address the vulnerabilities identified in CVE-2023-44487.
2. **Update google.golang.org/grpc**:
   Upgrade the `google.golang.org/grpc` dependency to version 1.58.3 or later to mitigate the risks associated with its current outdated version.
3. **Review and Monitor Dependencies**:
   Implement a systematic process for regularly reviewing and updating

dependencies to ensure security and compatibility. This may involve automated tooling to flag outdated or vulnerable dependencies.

Addressing these updates is essential for maintaining the security integrity of the Go-Ethereum project and protecting it from potential exploits.

## F-2024-2042 - Potential Exposure of Staking Node Keys - Low

**Description:** Staking nodes' keys are located in the **staking/mainnet** directory of the private repository for the Dione protocol's Odyssey chain. It has not been verified if these are actual mainnet staking keys and certificates. However, if they are genuine, this represents a critical security risk, as storing sensitive keys in the repository can lead to unauthorized access, even if the repository is private.

**Assets:**

- Odyssey chain
- Assets & Incentives

**Status:** Fixed

### Classification

**Severity:** Low

**Impact:** 2/5

**Likelihood:** 2/5

### Recommendations

**Remediation:**

1. **Verify Key Authenticity**: Determine whether the keys in the **staking/mainnet** directory are indeed real mainnet staking keys and certificates.
2. **Secure Key Storage**: If the keys are verified as real, immediately remove them from the repository and store them securely using a secrets management system that restricts access and is auditable.
3. **Review Access Permissions**: Conduct a thorough access control audit for the repository to ensure that only necessary personnel have access, minimizing the risk of insider threats.

### Evidences

**Reproduce:** https://github.com/DioneProtocol/odysseygo/commit/3225b2ede222bdcc4b7c67bc24ff7a2b6479e0f7

## Observation Details

### F-2024-2037 - Comprehensive Code Quality Issues in Coreth and Odysseygo - Info

**Description:**

Multiple code quality issues have been identified in the Coreth and Odysseygo repositories. These issues range from inconsistent documentation and compilation errors to a lack of Continuous Integration (CI) pipelines, impacting the overall maintainability and reliability of these projects.

1. Inconsistent Documentation
In the Odysseygo documentation, repository names are outdated and do not reflect the current naming conventions of Coreth and Odysseygo. An example is found in the following snippet:
```sh
cd $GOPATH/src/github.com/DioneProtocol/odysseygo
go mod edit -replace github.com/DioneProtocol/coreth=../coreth
```
Also links are outdated or misleading:
- https://docs.dioneprotocol.com/learn/platform-overview
- https://docs.dioneprotocol.com/build/references/coreth-arc20s
2. Compilation Errors in Coreth
- params/config.go:33:2: The “time” package is imported but not used.
- plugin/evm/export_tx.go:111:71: The constant OmegaChainID is undefined.
3. Compilation Errors in Odysseygo, here are some of them:
- In github.com/DioneProtocol/odysseygo/vms/proposervm, types *postForkBlock and *preForkBlock fail to implement required interfaces due to missing methods.
- In github.com/DioneProtocol/odysseygo/snow/networking/router, constants.PlatformChainID is undefined.
- Syntax error in github.com/DioneProtocol/odysseygo/vms/omegavm/reward/calculator.go.
4. Code Formatting in both repositories does not adhere to recognized best practices for formatting.
5. Broken and failing tests:
- There are broken and failing unit and end-to-end tests in both repositories, indicating potential issues in the codebase that need to be addressed.
6. Absence of CI pipeline:
Neither repository has a CI pipeline set up. This means that code issues such as broken builds, failing test suites, and linting problems are not being automatically detected and addressed.

**Assets:**

- Coreth fork review

- Avalanchego fork

**Status:** Fixed

## Recommendations

**Remediation:**

1. Update Documentation: Revise and update all documentation to reflect the current state for naming conventionsa and all external documentation links.
2. Fix Compilation Errors: Review and rectify all instances of compilation errors in both repositories.
3. Enforce Code Formatting Standards: Implement and enforce a code formatting standard across both repositories.
4. Repair Broken Tests: Examine and fix all broken or failing tests to ensure they are functioning as intended.
5. Establish CI Pipelines: Set up CI pipelines for both repositories to automatically detect and report issues related to the build process, testing, and code quality.
6. Addressing these issues will significantly improve the code quality, reliability, and maintainability of both the Coreth and Odysseygo projects.

## F-2024-2038 - E2E Suite: Interchain Transfer Failure on D-Chain - Info

**Description:** Funds transfer from the D-Chain to the A-Chain and the O-Chain failed due to an "invalid sender" error, indicating potential issues with transaction sender validation or wallet configuration. Output:

**Assets:**

- Odyssey chain
- Node Tests

**Status:** Fixed

### Recommendations

**Remediation:** **Review and Correct Sender Validation Logic**: Ensure the sender's address and authentication are correctly validated in the interchain workflow processes to prevent "invalid sender" errors.

### Evidences

**Issue details:**

```
[D-Chain] [Interchain Workflow]
/root/sfxdx__odysseychain/tests/e2e/describe.go:31
[It] should ensure that funds can be transferred from the D-Chain to
the A-Chain and the O-Chain
/root/sfxdx__odysseychain/tests/e2e/d/interchain_workflow.go:34

Begin Captured GinkgoWriter Output >>
STEP: initializing a new eth client 04/06/24 17:50:00.995
targeting node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeLeer5juM with URI: ht
tp://127.0.0.1:44327
initializing a new eth client for node NodeID-Nw33LEJEAcScfo2HiKvW3g
vEeLeer5juM with URI: http://127.0.0.1:44327
STEP: allocating a pre-funded key to send from and a recipient key t
o deliver to 04/06/24 17:50:00.995
allocated funded key(s): [PrivateKey-5F2ioX4tKWu2TzMmP4WabE2Jiu8CnbB
p2xFWSdgMWj2Bmr7Kp]
STEP: sending funds from one address to another on the D-Chain 04/06
/24 17:50:00.996
sending eth transaction with ID: 0x3285eb30e138efb18553d10566c02cd3a
11882de8590f599be0bb9e2d8aeb0c2
<< End Captured GinkgoWriter Output


Error Trace: /root/sfxdx__odysseychain/tests/e2e/e2e.go:215
/root/sfxdx__odysseychain/tests/e2e/d/interchain_workflow.go:71
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/core_dsl.go:535
/root/sfxdx__odysseychain/tests/e2e/d/interchain_workflow.go:50
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/internal/node.go:4
49
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/internal/suite.go:
750
/usr/local/go/src/runtime/asm_amd64.s:1695
Error: Received unexpected error:
invalid sender
```

## F-2024-2039 - E2E Suite: Interchain Transfer Failure on A-Chain - Info

**Description:** Funds transfer to the D-Chain and the O-Chain resulted in "insufficient funds," suggesting issues in balance management or transaction execution.

**Assets:**

- Odyssey chain
- Node Tests

**Status:** Fixed

### Recommendations

**Remediation:** **Audit Balance Management and Transaction Execution**: Investigate the balance management system and transaction execution flow, particularly for cross-chain transfers, to resolve "insufficient funds" errors.

### Evidences

**Issue details:**

```
[A-Chain] [Interchain Workflow]
/root/sfxdx__odysseychain/tests/e2e/describe.go:13
[It] should ensure that funds can be transferred from the A-Chain to
the D-Chain and the O-Chain
/root/sfxdx__odysseychain/tests/e2e/a/interchain_workflow.go:31

Begin Captured GinkgoWriter Output >>
targeting node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeLeer5juM with URI: ht
tp://127.0.0.1:44327
STEP: creating wallet with a funded key to send from and recipient k
ey to deliver to 04/06/24 17:50:00.997
allocated funded key(s): [PrivateKey-VjpGtyYvCscb81HbFjSkrvMVP66Aut6
HvVXXpdudpEdPV3kFe]
initializing a new wallet for node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeL
eer5juM with URI: http://127.0.0.1:44327
STEP: defining common configuration 04/06/24 17:50:01.005
STEP: sending funds from one address to another on the A-Chain 04/06
/24 17:50:01.005
issued transaction with ID: 2Tz4i1FP9pZjg3nxGdKBgzDC6xZKKRQZA4bwoBKB
dCMbZPxrYY
STEP: checking that the A-Chain recipient address has received the s
ent funds 04/06/24 17:50:01.114
STEP: exporting DIONE from the A-Chain to the D-Chain 04/06/24 17:50
:01.114
issued transaction with ID: 2LSpJW2QLK73Du1rLi3JbWXR8AtbqTSjxVqsuiBC
MeCVMWpbHX
STEP: initializing a new eth client 04/06/24 17:50:02.016
initializing a new eth client for node NodeID-Nw33LEJEAcScfo2HiKvW3g
vEeLeer5juM with URI: http://127.0.0.1:44327
STEP: importing DIONE from the A-Chain to the D-Chain 04/06/24 17:50
:02.016
<< End Captured GinkgoWriter Output


Error Trace: /root/sfxdx__odysseychain/tests/e2e/a/interchain_workfl
ow.go:116
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/core_dsl.go:535
/root/sfxdx__odysseychain/tests/e2e/a/interchain_workflow.go:109
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/internal/node.go:4
49
/root/go/pkg/mod/github.com/onsi/ginkgo/v2@v2.4.0/internal/suite.go:
```

```
750
/usr/local/go/src/runtime/asm_amd64.s:1695
Error: Received unexpected error:
insufficient funds
```

## F-2024-2040 - E2E Suite: Permissionless Subnet Operations on O-Chain - Info

**Description:** The O-Chain failed to properly handle subnet operations, yielding a "couldn't issue tx: not found" error, which may point to deficiencies in transaction broadcasting or asset management within the subnet.

**Assets:**

- Odyssey chain
- Node Tests

**Status:** Fixed

### Recommendations

**Remediation:** **Enhance Subnet Transaction Management**: Examine the permissionless subnet's transaction handling mechanisms, focusing on the accuracy and reliability of transaction issue and retrieval processes.

### Evidences

**Issue details:**

```
[O-Chain] [Permissionless Subnets]
/root/sfxdx__odysseychain/tests/e2e/describe.go:25
[It] subnets operations [xp, permissionless-subnets]
/root/sfxdx__odysseychain/tests/e2e/o/permissionless_subnets.go:30

Begin Captured GinkgoWriter Output >>
targeting node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeLeer5juM with URI: ht
tp://127.0.0.1:44327
allocated funded key(s): [PrivateKey-25zAFkgadP43YEf7SAJ1p956xkgTpAd
6jkXRwR7DsdJVfKERj3]
initializing a new wallet for node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeL
eer5juM with URI: http://127.0.0.1:44327
STEP: retrieving the node ID of a primary network validator 04/06/24
17:50:01.015
STEP: create a permissioned subnet 04/06/24 17:50:01.016
issued transaction with ID: yvCKuKVP9BswmXtiH5MRTid6Mc93Nd3ZTqvmm4yk
edvAqES65
STEP: create a custom asset for the permissionless subnet 04/06/24 1
7:50:02.017
issued transaction with ID: 2P92TqN8YA6WQqLJrsYt8sBC2RHoCQhrmw3T8xuC
xdH1vznUjV
STEP: Send 100 MegaDione of asset 2P92TqN8YA6WQqLJrsYt8sBC2RHoCQhrmw
3T8xuCxdH1vznUjV to the O-chain 04/06/24 17:50:03.019
issued transaction with ID: RwrqDmgfptHS6szLxYSRxhr5mDeoNmfrhUfGHXvy
YGKHc5quP
STEP: Import the 100 MegaDione of asset 2P92TqN8YA6WQqLJrsYt8sBC2RHo
CQhrmw3T8xuCxdH1vznUjV from the A-chain into the O-chain 04/06/24 17
:50:04.121
issued transaction with ID: Q8dmP4ep1AEkbVSD9S4zmxNoruqjV3eJdCSvTBHM
ZvGnjR6Ur
STEP: add permissionless validator 04/06/24 17:50:05.022
<< End Captured GinkgoWriter Output

Expected
<*fmt.wrapError | 0xc000143980>: {
msg: "failed to decode client response: couldn't issue tx: not found
",
```

```
err: <*json2.Error | 0xc000416450>{
Code: -32000,
Message: "couldn't issue tx: not found",
Data: nil,
},
}
to be nil
```

## F-2024-2041 - E2E Suite: Virtuous Transfer Transaction Failure on A-Chain - Info

**Description:** A virtuous transfer transaction for the DIONE asset failed due to a mismatch in expected and actual balances, hinting at potential problems in transaction processing or balance updating mechanisms.

**Assets:**

- Odyssey chain
- Node Tests

**Status:** Fixed

### Recommendations

**Remediation:** **Validate Transaction Processing and Balance Updating**: Thoroughly test the virtuous transfer transaction logic and balance updating routines to ensure they correctly reflect post-transaction states.

### Evidences

**Issue details:**

```
[A-Chain] [Virtuous Transfer Tx DIONE]
/root/sfxdx__odysseychain/tests/e2e/describe.go:19
[It] can issue a virtuous transfer tx for DIONE asset [a, virtuous-t
ransfer-tx-dione]
/root/sfxdx__odysseychain/tests/e2e/a/transfer/virtuous.go:39

Begin Captured StdOut/StdErr Output >>
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "KgTsTXPrEqji
EK5wEJhWskqjR6oZGysyb")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "N3CSRvcLY3LK
JZpM9B9HjpKMURNfw56Kg")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "3CHh7dYW8w4o
Wosm2UVqgSYHXaxEH7ohg")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "5r6v2v29BECf
dkChacAn4RaLemGEoNV5y")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "7vi7Vr6XEV7P
S6Fer6fGFwTyr6Vnv8xY4")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "796bq333nMai
XotNGTJRRM6TQfMS4ZJ8t")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "L6u69ggejAYE
6KW9vRhGNATLdf5TXiG4h")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "BoTywzgVEHnS
t44aGmgFnsrnp88Cu93o1")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "NWgsHBwkrqgv
AKmwPupZkfeWgGaG2ELN4")
CURRENT BALANCE 30000000000000000 DIONE (SHORT ADDRESS "M7eF4W8wgHRu
AEi2cvFt5Gpq6YSvQYXso")
===
TRANSFERRING

FROM ["KgTsTXPrEqjiEK5wEJhWskqjR6oZGysyb"]
SENDER CURRENT BALANCE : 30000000000000000 DIONE
SENDER NEW BALANCE (AFTER) : 26999999999000000 DIONE

TRANSFER AMOUNT FROM SENDER : 3000000000000000 DIONE

TO ["N3CSRvcLY3LKJZpM9B9HjpKMURNfw56Kg"]
```

```
RECEIVER CURRENT BALANCE : 30000000000000000 DIONE
RECEIVER NEW BALANCE (AFTER) : 33000000000000000 DIONE
===
<< End Captured StdOut/StdErr Output

Begin Captured GinkgoWriter Output >>
allocated funded key(s): [PrivateKey-Uybztc3mLqawGy89G1XLTDVWapgdBzs
YxB1kVGXkmqjdWMy4P PrivateKey-pVFXwxsRtdy3ryBwJiDUJkzEJycxQzoGV2GqoZ
M3eUvMMWZoR PrivateKey-YQUiFFMdkHzQcMd7uAY1k4cJKcNC5yC6dshMwTsJzpzwU
CLXC PrivateKey-2Gg7Q3oSLVGA2j12mGPQxhMjbVjJMTwSSFLSDvazn7JBgdzX4u P
rivateKey-FmytfJq5AQys4R8McC98Qfk8YjrJDjkuahCRNnWWXuzXfJzcJ PrivateK
ey-2GTfBXnMBfs939dXhsNJ4wGnCGKGGgY9Uw74QDDHMSbRzm2xNQ PrivateKey-2tX
jfNe2KDB1vK11pk4DeHv6yEVzGEpQDwizoWuCzwT5srocDi PrivateKey-ktACpJYX7
osQms5HFHGJHG41KrEdnYqBax8eYFmfMusVVooAX PrivateKey-yKf3fUgKJjGxmLN1
SRTo5i9ZXeZDEPKH9thGbu1fuUNDT87bC PrivateKey-mLZPpaiRusGY22uBrF2MgPA
tYbBzPpdApV5scRSvpxudFmtNV]

---
[ROUND #00]:
targeting node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeLeer5juM with URI: ht
tp://127.0.0.1:44327
initializing a new wallet for node NodeID-Nw33LEJEAcScfo2HiKvW3gvEeL
eer5juM with URI: http://127.0.0.1:44327
metrics at "http://127.0.0.1:41649": map[]
metrics at "http://127.0.0.1:44327": map[]
STEP: A-Chain transfer with wrong amount must fail 04/06/24 17:52:12
.133
issued transaction with ID: 2cqjNxVbPDEN15bc1aUy89MPB7ugr7KvKcG7TfRV
FxYdFhtbc2
first wallet balance: 26999999999000000
second wallet balance: 33000000000000000
<< End Captured GinkgoWriter Output

Expected
<float64>: 0
to equal
<float64>: 1
```

## F-2024-2043 - Test Coverage and High Failure Rate in Genesis Package - Info

**Description:**

Out of the total tests executed, 31 failed, 4077 passed, and 2 were ignored.

Notably, 29 of the failed tests are under the **genesis** package.
Here are specific tests in the genesis package that are problematic, such as:

- TestGenesisFromFile
- TestGenesisFromFlag
- TestGenesis
- TestVMGenesis
- TestDioneAssetID

**Assets:**

- Odyssey chain
- VM
- Node Tests

**Status:** Fixed

### Recommendations

**Remediation:**

**Prioritize Fixing Genesis-Related Tests**: Focus on resolving the failures in the **genesis** package, starting with the identified tests (**TestGenesisFromFile**, **TestGenesisFromFlag**, **TestGenesis**, **TestVMGenesis**, **TestDioneAssetID**). This may involve reviewing and updating the logic related to the genesis block's configuration and initialization processes.

### Evidences

#### Coverage data

**Issue details:**

Private link to coverage data:
https://drive.google.com/file/d/1CluYmA9Lj6Xljkv5awzBdM7CTWJP8YNP/view?usp=drive_link
Failed tests are not included.

# Appendix 1. Severity Definitions

| Severity | Description |
| --- | --- |
| Critical | Vulnerabilities that can lead to a complete breakdown of the blockchain network's security, privacy, integrity, or availability fall under this category. They can disrupt the consensus mechanism, enabling a malicious entity to take control of the majority of nodes or facilitate 51% attacks. In addition, issues that could lead to widespread crashing of nodes, leading to a complete breakdown or significant halt of the network, are also considered critical along with issues that can lead to a massive theft of assets. Immediate attention and mitigation are required. |
| High | High severity vulnerabilities are those that do not immediately risk the complete security or integrity of the network but can cause substantial harm. These are issues that could cause the crashing of several nodes, leading to temporary disruption of the network, or could manipulate the consensus mechanism to a certain extent, but not enough to execute a 51% attack. Partial breaches of privacy, unauthorized but limited access to sensitive information, and affecting the reliable execution of smart contracts also fall under this category. |
| Medium | Medium severity vulnerabilities could negatively affect the blockchain protocol but are usually not capable of causing catastrophic damage. These could include vulnerabilities that allow minor breaches of user privacy, can slow down transaction processing, or can lead to relatively small financial losses. It may be possible to exploit these vulnerabilities under specific circumstances, or they may require a high level of access to exploit effectively. |
| Low | Low severity vulnerabilities are minor flaws in the blockchain protocol that might not have a direct impact on security but could cause minor inefficiencies in transaction processing or slight delays in block propagation. They might include vulnerabilities that allow attackers to cause nuisance-level disruptions or are only exploitable under extremely rare and specific conditions. These vulnerabilities should be corrected but do not represent an immediate threat to the system. |

# Appendix 2. Scope

The scope of the project includes the following components from the provided repository:

| Scope Details | |
|---|---|
| Repository | https://github.com/DioneProtocol/odysseygo |
| Commit | b44df2531bd9b33cbc8e778f64f1f8cfb5d8c602 |
| Repository | https://github.com/DioneProtocol/coreth |
| Commit | f9d2ba69b402b761854f09413cb270110e65333c |
| Repository | https://github.com/DioneProtocol/sfxdx__orion-sc |
| Commit | 5755426b9d309993d96b27c6eb4240d36249deb9 |
| Whitepaper | https://www.dioneprotocol.com/whitepaper.pdf |
| Requirements | DP-Blockchain Customizations-110324-102732.pdf |
| Technical Requirements | DP-Blockchain Customizations-110324-102732.pdf |