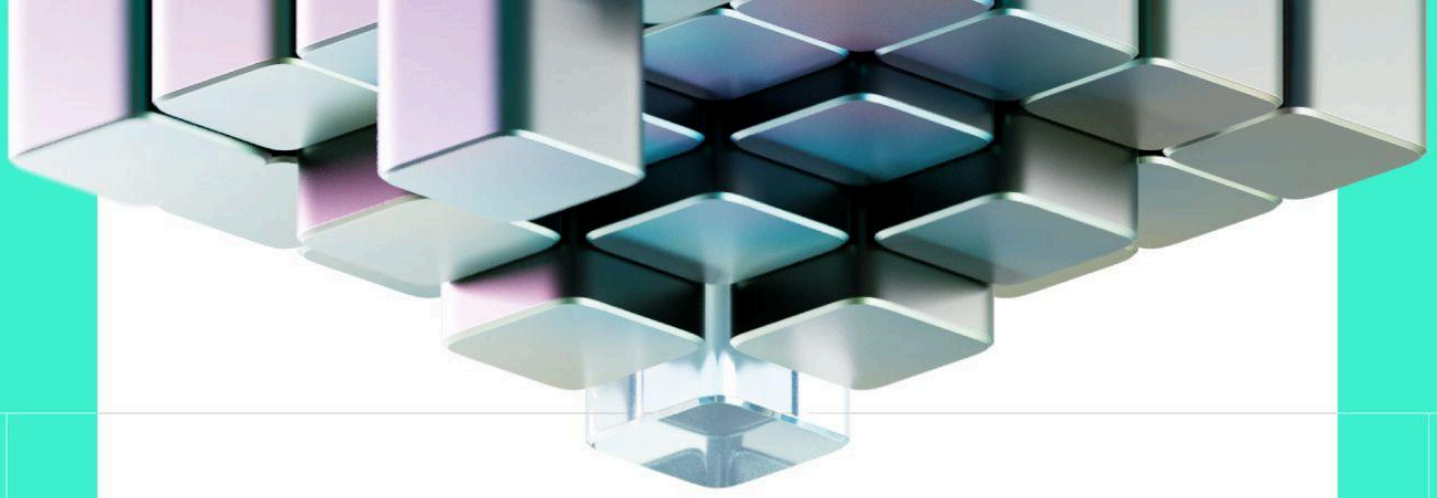
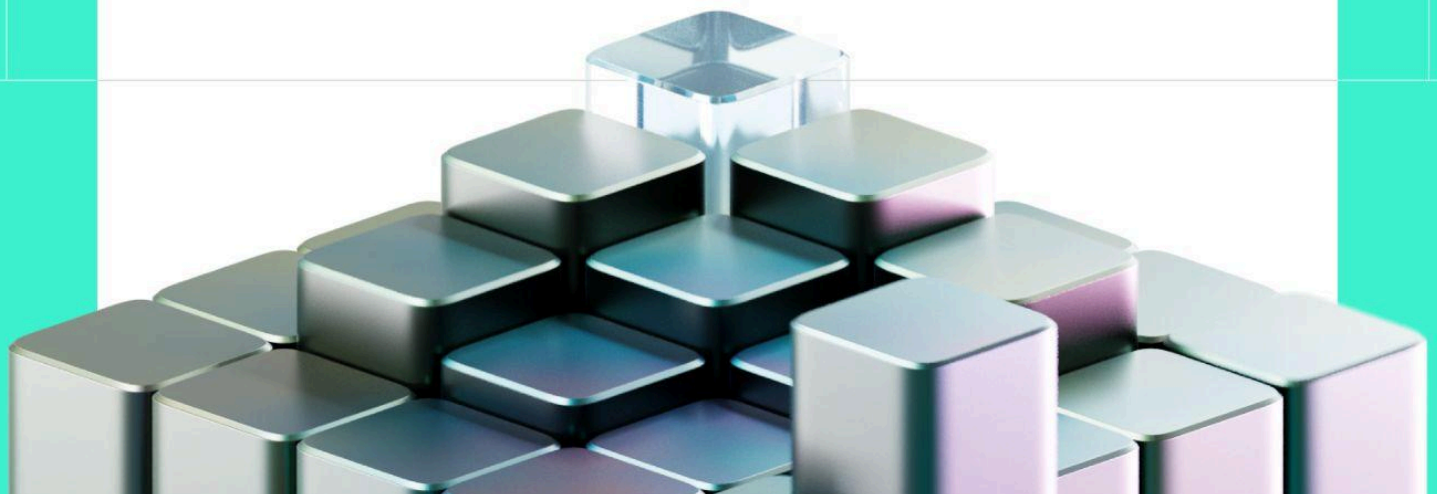


HACKEN

SMART CONTRACT PRE-AUDIT REPORT

REClosure| 10 January 2024



Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Executive Overview

Hacken's Smart Contract Audit Methodology

At Hacken, our Smart Contracts Audit and Analysis Methodology is a robust response to customer demands, leveraging years of experience in smart contracts audits. The objective is to elucidate the audit process, emphasizing its significance in enhancing smart contract development quality and safeguarding Hacken customers' funds. The surge in crypto adoption, exemplified by a $2.64 trillion market capitalization and DeFi surpassing $150 billion, underscores the need for heightened security. In 2021 alone, crypto thefts amounted to $681 million, with DeFi being a primary target.

The Value of Smart Contracts Audit: Smart contracts, once deployed, are immutable, necessitating proactive security measures. Our methodology empowers developers and project owners by:

1. **Ensuring Development Alignment:**
    - Confirming development aligns with functional requirements.
    - Providing clarity on pre, during, and post-audit steps.
    - Understanding the security scoring model.
2. **Guidance for Developers:**
    - Offering insights into the audit process and its phases.
    - Furnishing secure development recommendations.
    - Describing common mistakes and best practices.

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Audit Strategy and Framework

Our Methodology and Roadmap

Our Smart Contracts Audit and Analysis Methodology stands as a cornerstone for secure and high-quality smart contract development, empowering projects to thrive in the dynamic landscape of crypto adoption.

1. **Preparation:**
    - Importance of Preparation: Vital for thorough project review.
    - Key Steps:
        - Prepare functional and technical project requirements.
        - Configure the development environment.
        - Develop comprehensive unit tests.
        - Ensure adherence to code style and best practices.
    - Pre-audit: Helps customers understand audit preparation.
2. **Code Review and Analysis (Testing):**
    - Overall Review: High-level understanding of code structure.
    - Automated Tools Scan: Uses tools like Slither, Mythril, Solgraph (Solidity), Clippy, Cargo-udeps, Cargo-audit (Rust)...
    - Funds and Data Flow Diagrams: Visualizes contract states and interactions.
    - Line-to-Line Review: Thorough examination for issues, adherence to style guides, and best practices.
    - Unit Tests: Emulates contract usage by multiple users to prevent vulnerabilities.
3. **Reporting:** Comprehensive documentation of findings, recommendations, and security scoring.

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Scope of Engagement

At the heart of our audit lies a meticulous examination of your project's codebase. We begin by defining the scope of our engagement, providing clarity on the repositories and commits that will be subject to our assessment.

In-scope Repositories and Commits

Repository:	https://git.alisio-computing.eu/reclosure/security-token/-/tree/cc5d9dd423625e9169ddc3342350675d376de6d1
Commit Hash:	cc5d9dd423625e9169ddc3342350675d376de6d1

Repository

It is recommended to centralize the source code, tests, and development environment intended for the audit within a unified repository.

Current Status : Repository management is sufficient.

Commit

It is preferable that there are no subsequent commits beyond the agreed-upon state when initiating the audit process, indicating the completion of development. In the event that such commits exist, it is the customer's responsibility to determine whether to proceed with the most recent commit or the commit provided at the time of the agreement. **HACKEN bears no responsibility for identifying issues beyond the defined scope**.

Current Status : The provided commit is the latest.

Contracts

Contracts, abstract contracts and libraries are counted in the total number of lines of codes (LoC).

Total LoC : 815

File Path & SHA-3 Hash	LoC
File: src/DSIP.sol SHA3: 263b4eb361698c713ed401675205b2a457d5357ede65f3fbe3f6ebbf31dfb19f	322
File: src/IdentityManager.sol SHA3: 556f9efef3506c558f79953cc269bbdfce44e4e79a43b69691cc4ccfc7266018	18
File: src/OrderBook.sol SHA3: 962b78f33cc515aa46f7d5bdbdcc0cb4be6b1f1075bd116ceb44ecf25729ac10	118
File: src/extensions/DividendManager.sol SHA3: 74bb92520f339f22270b54b1223a42c6c8c2f352b35d1d2bb1650e326423a5d1	129
File: src/extensions/FeeManager.sol SHA3: 6dc94bf7ee3e8c8b8253186a16d3dda9702c3ae3c4d6494174674432a9c27d50	156
File: src/extensions/LockManager.sol SHA3: 02e92f390d16dbab9fedfa3a092f161b7ecbd3f260f2c665a225e80cad9bdb6b	36
File: src/extensions/SaleManager.sol SHA3: 7c0db7a91e9f02d4cc0def33f5911a292ac8339b43f0fc1baac94cbbcdba0c0d	36

Interfaces

Interfaces are not counted in the total number of lines of codes (LoC).

File Path & SHA-3 Hash	LoC
File: src/interfaces/IDSIP.sol SHA3: 8becd4d0d175e3c4e6e8b8d69067240771f1ee8dcf64fcc29e9e0567951503e8	66
File: src/interfaces/IIdentityManager.sol SHA3: 18bb02902ff13b7ac7120b4d6d55bbf3fd173575164e7e2abf3c7d57bfafe251	7
File: src/interfaces/IOrderBook.sol SHA3: ce269b36570d482d91a4ea6fe7ef05f643c0c6142929752c8e2d6ea9c62e7ff6	39

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Missing Dependencies

There are no missing dependencies.

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Project pre-analysis

Platform:	**EVM**
Language:	**Solidity**

Environment Configuration & Deployment Instructions

Missing deployment instructions will require longer time and effort as it will make it difficult for the auditors to understand and examine the project. This will be reflected in the audit fee and duration. In addition to these, a misconfigured environment will reduce the effectiveness of the audit.

Compilation Status: The code does not compile.

Deployment Status: There are no deployment instructions and the code can not be deployed

Test Coverage

Testing is a powerful tool that allows verifying that implementation is compliant with requirements. Failures and boundary value cases should be checked for maximum performance.

Tests should be configured to run on the project environment without the necessity to start any third-party tools like local Ethereum node, etc. In case of uncommon repository configuration, instructions on how to run the test coverage measurement should be provided.

Status : The current test coverage is **0**% (Branch). Tests can not be run due to the lack of instructions for finding coverage.

Implementation Details

The following elements, when present in an audit, require additional efforts for auditors to accurately identify potential vulnerabilities :

Metric	Status
The project can send/receive funds	Y
Assembly Usage	N
Low-Level Calls	N
Delegate Calls	N
Hash Functions	N
External Interactions	Y
Complex Formula for Fees/Interest/Rewards/Logic	Y

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Documentation

Functional Requirements

For any project, the availability of transparently documented functional requirements is essential. These specifications elucidate the project's objectives and the methods employed to achieve them. Detailed explanations of each smart contract entry point significantly enhance the efficiency of code examination.

Inadequate functional requirements may lead to the identification of numerous false-positive findings. Furthermore, the absence of such requirements presents challenges for auditors in understanding the project's business logic, assumptions, formulas, requirements, and features, potentially resulting in overlooking certain logic bugs. Additionally, the omission of functional requirements adversely impacts the documentation quality score of the audit report.

Functional Requirements link: README.md

Current State: The functional requirements are partially provided.

Technical Requirements

The absence of technical requirements poses challenges for auditors in comprehending the project's architectural logic, key functions, infrastructure, assumptions, and utilized technologies. This, in turn, complicates the identification of specific logic bugs that may be present in the project.

To facilitate developers’ and auditors' involvement in the project, it is imperative for any project to have well-documented technical requirements. This

documentation should cover the mathematical formulas utilized within the project. Insufficient technical requirements can result in longer estimation periods and higher charges.

Technical Requirements link: README.md

Current State: The technical requirements are provided.

NatSpecs

NatSpecs are an indispensable component of a comprehensive audit process, ensuring a thorough understanding of a project's nature and functionality. Clear and accessible NatSpecs serve as a foundational guide, shedding light on the project's objectives, operational pathways, and business logic.

Their absence introduces ambiguity, making it arduous for auditors to perform a thorough examination and potentially leaving critical issues undiscovered. Embracing NatSpecs ensures clarity, efficiency, and accuracy in the audit, ultimately contributing to the project's overall success.

Current State: The NatSpecs of the project are missing

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Action Points

- **Provide deployment instructions.**
- **Provide instructions to measure the test coverage of the project.**
- **Improve functional requirements with the following :**
    - **Inform the users about the different roles and authorizations**
- **Add the NatSpecs.**

Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Deliverables

The deliverable of the smart contracts audit review is a detailed report that contains:

01 An executive summary containing a brief description of the project.

02 Actions performed during the audit (methodology used).

03 Scores with detailed descriptions from 0 to 10 in Documentation Quality, Code Quality, Test Coverage and Security Score.

04 Findings that represent security issues with description and mitigation recommendations.

05 Violations of best practices that don’t have a security impact.

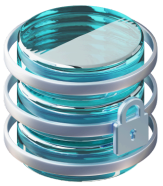
Hacken OU
Parda 4, Kesklinn, Tallinn
10151 Harju Maakond, Eesti
Kesklinna, Estonia

Hacken: Your Strategic Advantage

At Hacken, we understand that the cornerstone of any thriving blockchain ecosystem is unassailable security. Our approach is designed to instill customer confidence through a suite of high-level services, ensuring that stakeholders can operate in a secure and stable environment. Our team's deep-rooted expertise in blockchain technology and cybersecurity positions us as a leader in the field, providing strategic advantages to our partners.

Choosing Hacken means opting for a partnership that prioritizes your security needs with tailored, cutting-edge solutions. Our vigilant approach to staying ahead of the curve in cybersecurity trends ensures that your ecosystem is equipped to face the challenges of today and tomorrow. With Hacken, you gain more than a service provider; you gain an ally committed to fortifying your digital landscape.

Strengthen customer confidence



One of the world's first audit of digital asset security

Increase competitiveness

6+	150+	180+	1,000+	$20B+
Years of expertise	Team members	Partners	Clients	Daily Secured