# TON Smart Contracts Vulnerabilities

Mishuris Matvey

Open Network (TON) - это высокопроизводительная блокчейн-платформа, разработанная для обеспечения масштабируемости и эффективности, использующая модель асинхронного выполнения и многоуровневую архитектуру.

Хотя дизайн TON обладает значительными преимуществами, он также создает уникальные проблемы для разработки смарт-контрактов и обеспечения безопасности.

Платформа поддерживает широкий спектр децентрализованных приложений, смарт-контрактов и сервисов микроплатежей, основанных на собственной криптовалюте Toncoin, которая облегчает обработку транзакций и обеспечивает безопасность сети.

TON

# Цель

Разработать инструмент для поиска и устранения уязвимостей в TON smart contracts

TON

# From Paradigm Shift to Audit Rift: Exploring Vulnerabilities and Audit Tips for TON Smart Contracts

TON

- Поиск и анализ существующих уязвимостей на основе доступных аудиторских отчётов, публичных репозиториев (GitHub)

- Классификация найденных уязвимостей

- Моделирование типичных сценариев возникновения выявленных уязвимостей разного рода

- Разработка стратегий по детектированию, устранению и предотвращению ошибок

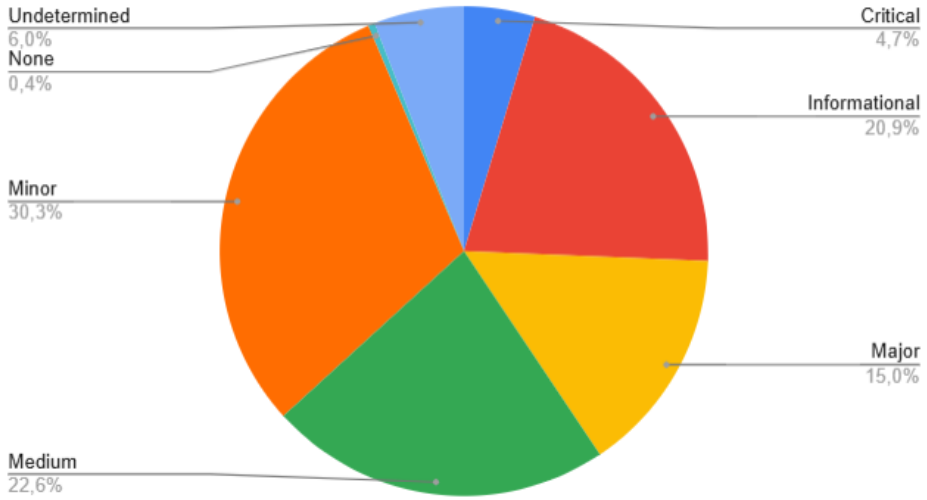
TON

- Одна из задач заключается в поиске и систематизации существующих ошибок на основе доступных ресурсов.

- На данный момент найдено более 30 аудиторских отчётов.

- В общей сложности было зарегистрировано более 200 уязвимостей.

| Security Level | Vulnerability Count |
|---|---|
| Critical | 11 |
| Major | 35 |
| Medium | 53 |
| Low | 71 |
| Informational | 49 |
| Undetermined | 14 |
| **Total** | 233 |




TON

# Все уязвимости были представлены в виде таблицы для дальнейшего анализа

| A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Project | Auditor | Date | Language | Security Level | Status | Vulnerability details | Type | Subtype | Subsubtype | Preliminary Type | Comment for Checklist |
| TonUP | TonBit | 2023-05 | Tact | Minor | Fixed | Incorrect event emit in SetTokenWalletAddress and SetUpWalletAddress functions. | Common Errors | Logical Errors | NA | Common Errors | Use correct event emitters to ensure accurate logging and transparency. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Minor | Acknowledged | Missing validation for `vesting_total_duration` in `locker.load_data()`. Consider checking `vesting_total_duration > 0`. | Contract Design | Input Data Processing | Input Data Processing | Input Data Processing | Ensure proper validation of input data as per TON Audit Guide. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Minor | Acknowledged | Time constraints on rewards and deposits can lead to unfair distribution. Consider implementing a shorter time limit for reward distribution. | Common Errors | Logical Errors | NA | Logical Errors | Check for logical errors and edge cases in time-based logic. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Minor | Acknowledged | Potential storage fee insufficiency in `locker_bill` contract. Consider limiting `vesting_start_time + vesting_total_duration`. | Gas Control | Moderate Handling of | Calculation of Costs | Gas Control | Ensure accurate gas control and storage fee management. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Minor | Acknowledged | Overflow and division by zero errors in edge cases. Consider adding validations for `unlock_period` | Common Errors | Logical Errors | NA | Logical Errors | `total_coins_locked` |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Minor | Acknowledged | Risk of capital inefficiency in reward-lacking deposits. Consider preventing deposits if no reward is added. | Common Errors | Logical Errors | NA | Logical Errors | Ensure proper handling of deposits and rewards. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Informational | Acknowledged | Inefficiency in reward reservation within the `locker` contract. Consider limiting reserved TON by checking the balance first. | Gas Control | Moderate Handling of | Calculation of Costs | Gas Control | Optimize gas usage and resource management. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Informational | Acknowledged | Redundant checks of immutable values in `load_data()`. Consider executing checks only once. | Best Practices | Code Review | NA | Code Efficiency | Improve code efficiency by reducing redundant operations. |
| TON Locker Contract | Quantstamp | 2023-07 | FunC | Informational | Acknowledged | Lack of documentation for minor business logic. Consider documenting reward and deposit handling in `README.md`. | Best Practices | Documentation | NA | Documentation | Ensure comprehensive documentation of business logic. |

Детализация классификации позволила разделить все уязвимости на подсекции, выделив тем самым основные типы ошибок.

- Vulnerabilities 233
  - Contract Design 76
    - Authorization Checks 25
    - Contract Design and Centralization 19
    - External Message Handling 1
    - Input Data Processing 15
    - Logical Errors 8
    - Message Generation and Handling 2
    - Partial Execution of Transactions 6
  - Asynchronous Execution 6
    - Key Considerations 6
  - Common Errors 92
    - Bounced Message Handlers 3
    - Carry-Value Pattern 1
    - Exit Codes 2
    - Logical Errors 70
    - Parsing and Serialization 6
    - Public Nature of Blockchain 2
    - Replay Attack 3
    - Restrictions on Data Recording 1
    - Sending Messages from Loops 1
    - Smart Contract Update 2
    - Smart Contract Updates 1
  - Gas Control 18
    - Moderate Handling of Gas 18
  - Random Number Generation in TON 1
    - Safe Randomness Generation 1
  - Possible Errors in FunC 4
    - Function Modifiers 1
    - Modifying Variables 2
    - Storage Management 1
  - Best Practices 36
    - Code Review 7
    - Compliance with Standards 5
    - Documentation 23
    - Magic Numbers-Flags-and Constants 1

TON

# Contract Design

Стандартные ошибки в коде смарт-контрактов:

- Необработанные исключения
- Дублирование кода
- Логические ошибки
- Неэффективные алгоритмы

TON

# Gas Control

Gas в контексте блокчейна - это комиссия за работу вычислительных ресурсов TVM.

Ошибки типа "Gas Control" связаны с неправильным расчётом комиссии за проведение транзакции.

TON

# BugMagnifier: TON Transaction Simulator to Reveal Smart Contract Vulnerabilities

TON

- Создание инструмента для моделирования и управления порядком транзакций в TON Blockchain

- Разработка серии смарт-контрактов, которые намеренно включают конкретные уязвимости

- Тестирование инструмента с помощью подготовленных контрактов с целью выявления конкретных типов ошибок

TON

- Работа требует написания смарт-контрактов на FunC и последующего анализа возникающих уязвимостей.

- В данный момент ведётся работа по изучению технической документации.

- Разработка смарт-контрактов.

TON

- Для анализа и работы с уязвимостями необходимо иметь возможность локально тестировать смарт-контракты.

- Для этих целей ведётся поиск наиболее релевантных инструментов:

- Blueprint
- Sandbox
- TVM linker
- TON Testnet

TON

# Ссылки:

- https://docs.ton.org/v3/documentation/smart-contracts/func/docs/functions
- https://github.com/ton-org/sandbox
- https://test.ton.org/tblkch.pdf
- https://github.com/ton-org/blueprint?tab=readme-ov-file#contract-development
- https://tonbit.xyz/reports/TonUP-Smart-Contract-Final-Audit-Report.pdf
- https://certificate.quantstamp.com/full/ton-locker-contract/6872997f-1110-45cc-b70f-2a4cd639da1f/index.html

TON