InterFi NETWORK

SMART CONTRACT AUDIT

interfinetwork

hello@interfi.network

https://interfi.network

PREPARED FOR

DRAGON9 (TON)

INTRODUCTION

Auditing Firm	InterFi Network
Client Firm	Dragon9
Methodology	Automated Analysis, Manual Code Review
Language	`func`
Standard	`jetton`
Contract	EQB5k5SnuNwiGU1uf6oXkf4haqodc1nk7S7tibUU-kWZksLg
Blockchain	The Open Network (TON)
Ownership	Revoked - UQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJKZ
Website	http://dragon9.vip/
Telegram	https://t.me/dragon9vip
Medium	https://medium.com/@dragon9vip
YouTube	https://www.youtube.com/@Dragon9VIP
X (Twitter)	https://x.com/Dragon9vip
Report Date	November 04, 2024

ℹ️ Verify the authenticity of this report on our website: https://www.github.com/interfinetwork

EXECUTIVE SUMMARY

InterFi has performed the automated and manual analysis of Ton contracts. Ton contracts were reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical 🔴	Major 🟠	Medium 🟡	Minor 🟢	Unknown 🟤
Open	0	0	1	1	0
Acknowledged	0	0	1	1	1
Resolved	0	1	0	5	0
Important Functions	`recv_internal(), mint_tokens(), burn_notification(), load_data(), save_data(),get_wallet_address()`				
Owner Privileges	Revoked 🟢 - UQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJKZ				

⚠ Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.

⚠ There is no public evidence of KYC verification through any external or recognized KYC services. As of this token audit, neither presale process nor associated dApp has been subjected to an external audit by a recognized third-party security firm. Given the lack of external KYC verification and absence of third-party audits, potential users are advised to exercise caution and perform thorough due diligence before participating in project or presale.

TABLE OF CONTENTS

SCOPE OF WORK

InterFi was consulted by Dragon9 to conduct the smart contract audit of their Ton source codes. The audit scope of work is strictly limited to mentioned func file(s) only:

- `jetton-minter.fc`

ℹ️ If source codes are not deployed on the main net, they can be modified or altered before main-net deployment. Verify the contract's deployment status below:

Public Contract Link	
https://tonscan.org/jetton/EQB5k5SnuNwiGU1uf6oXkf4haqodc1nk7S7tibUU-kWZksLg#source	
Contract Name	`jetton-minter.fc`
Compiler Version	`0.3.0`

AUDIT METHODOLOGY

For a comprehensive audit of a smart contract deployed on the TON blockchain, we follow a structured audit process that includes automated and manual analyses to identify common vulnerabilities, centralized exploits, and potential issues. Below is the detailed audit process, adapted specifically for FunC and Fift contracts on the TON platform. Here's a brief overview of InterFi's auditing process and methodology:

CONNECT

- The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.

AUDIT

- Automated analysis is performed to detect common smart contract vulnerabilities specific to the TON blockchain. We may use the following tools and frameworks for automated analysis:
  - TON Compiler for compiling and simulating interactions.
  - TON SDK Tools for automated vulnerability detection.
  - Smart Contract Validators integrated with the TON VM to simulate contract behaviors.
- Blockchain Simulations to detect operational risks, such as message forwarding errors, gas inefficiencies, or potential DoS (Denial of Service) attacks.
- A manual line-by-line audit is crucial for identifying vulnerabilities and potential exploits that automated tools might miss.

Centralized Exploits	○ Token Supply Manipulation ○ Access Control and Authorization ○ Ownership Control ○ Message Handling ○ Stop and Pause Trading

	○ Asset Manipulation
Custom Vulnerabilities Checks	○ Integer Overflow ○ Lack of Arbitrary Limits ○ Gas Optimization ○ Re-entrancy ○ Third-Party Dependencies ○ Typographical Errors ○ Requirement Violations ○ Message Forwarding Fees ○ Code Style Violations ○ Message Queue Issues ○ Race Conditions: ○ Gas Limit Constraints ○ Mutable Token Metadata ○ Language-Specific Warnings ○ Compiler-Specific Warnings

REPORT

- The auditing team provides a preliminary report specifying all the checks which have been performed and the findings thereof. The client's development team reviews the report and makes amendments to contract codes.
- The auditing team provides the final comprehensive report with open and unresolved issues.

PUBLISH

- The client may use the audit report internally or disclose it publicly.

ℹ️ It is important to note that there is no pass or fail in the audit, it is recommended to view the audit as an unbiased assessment of the safety of TON platform contracts.

RISK CATEGORIES

A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized:

Risk Type	Definition
Critical 🔴	These risks pose immediate and severe threats, such as asset theft, data manipulation, or complete loss of contract functionality. They are often easy to exploit and can lead to significant, irreparable damage. Immediate fix is required.
Major 🟠	These risks can significantly impact code performance and security, and they may indirectly lead to asset theft and data loss. They can allow unauthorized access or manipulation of sensitive functions if exploited. Fixing these risks are important.
Medium 🟡	These risks may create attack vectors under certain conditions. They may enable minor unauthorized actions or lead to inefficiencies that can be exploited indirectly to escalate privileges or impact functionality over time.
Minor 🟢	These risks may include inefficiencies, lack of optimizations, code-style violations. These should be addressed to enhance overall code quality and maintainability.
Unknown 🟤	These risks pose uncertain severity to the contract or those who interact with it. Immediate fix is required to mitigate risk uncertainty.

All statuses which are identified in the audit report are categorized here:

Status Type	Definition
Open	Risks are open.
Acknowledged	Risks are acknowledged, but not fixed.
Resolved	Risks are acknowledged and fixed.

OWNER PRIVILEGES

Centralization risk is a significant concern in the TON blockchain ecosystem. When a smart contract has a privileged role, the centralization-related risk increases, potentially leading to loss of control over assets or manipulation of the contract.

There are legitimate reasons for having privileged roles in TON contracts:

- Privileged roles can be granted the ability to pause `recv_internal()` the contract during external threats or attacks.
- Privileged roles may use functions like `add_user()` or `exclude_user()` to manage wallet addresses, user permissions, or transaction limits, which is useful for operational tasks such as presales or exchange listings.

However, authorizing privileged roles to an externally-controlled account (EOA) can be risky, as this makes the contract vulnerable to centralized control.

RECOMMENDATION:

To reduce centralization-related risks, the following best practices can be adopted:

- Private key of a privileged role must be carefully protected to avoid potential hacks or loss of control. This is especially critical in TON's asynchronous environment, where messages and access controls are spread across different components.
- Privileged roles should be shared across multi-signature wallets. This means multiple trusted parties must approve critical actions, reducing the risk of a single point of failure or central control.
- Once the contract has been deployed and is stable, consider renouncing ownership and revoking privileged roles. This eliminates the risk of centralized control altogether.

ℹ️ Understand the project's initial asset distribution. Assets in the liquidity pools should be locked. Assets outside the liquidity pools should be locked with a release schedule.

AUTOMATED REVIEW

Function	Importance	Definition
`recv_internal()`	🔴	Central function for processing incoming messages (minting, burning, providing wallet addresses, changing admin/content)
`mint_tokens()`	🟠	Handles the minting of new tokens, affecting the total supply
`load_data()`	🟠	Responsible for loading the contract's state, impacting total supply, admin, and content
`save_data()`	🟠	Responsible for saving the contract's state, impacting total supply, admin, and content
`get_jetton_data()`	🟢	Returns the wallet address for a given owner (state query)
`get_wallet_address()`	🟢	Important for user interaction but no direct impact on core contract behavior

MANUAL REVIEW

Identifier	Definition	Severity
CEN-01	Owner privileges	Major 🟠

Smart contract relies heavily on comparing the `sender_address` with the `admin_address` for access control.

`op == 3` and `op == 4` operations only rely on a comparison, but if the `admin_address` were somehow tampered with, this can be exploited.

```
if (op == 3 || op == 4) { ;; change admin or change content
    throw_unless(73, equal_slices(sender_address, admin_address));
    ...
}
```


RECOMMENDATION

Securing private keys or access credentials of deployers, contract owners, operators, and other roles with privileged access is crucial to prevent single points of failure that can compromise contract security.

- Use of multi-signature wallets – These wallets require multiple authorizations to execute sensitive contract functions, reducing the risk associated with single-party control.
- Revoke ownership - Once the contract has been deployed, consider renouncing ownership and revoking privileged roles. This eliminates the risk of centralized control altogether.

RESOLUTION

Dragon9 team has revoked smart contract ownership.

Identifier	Definition	Severity
CEN-02	Initial token allocation	Medium 🟡

Upon deployment, all initially minted $DAN tokens are transferred to the contract deployer. It could be an issue as the deployer can distribute tokens without consulting the community.

```
Total supply – 9000 DAN
```


RECOMMENDATION

Establish transparent tokenomics model that involves community input in the decision-making process regarding token allocation.

ACKNOWLEDGEMENT

Dragon9 team has clarified that initial token allocation will adhere strictly to pre-determined tokenomics outlined in project documentation.

Identifier	Definition	Severity
CEN-03	Lack of token immutability	Minor

Smart contract allows the content (metadata) to be changed via `op == 4`.

RECOMMENDATION

Implement safeguards or completely remove the op == 4 operation to ensure that token metadata remains immutable

RESOLUTION

Dragon9 team has revoked smart contract ownership, hence, content (metadata) cannot be altered by any means after ownership revocation.

Identifier	Definition	Severity
LOG-01	Gas consumption	Minor 🟢

Smart contract enforces a minimum message value

```
if (op == op::provide_wallet_address()) {
    throw_unless(75, msg_value > fwd_fee + const::provide_address_gas_consumption());
```

for wallet address provision. If gas consumption estimates are inaccurate, the contract can reject legitimate transactions or consume excess gas.

RECOMMENDATION

Ensure that gas consumption estimation is accurate for various transaction sizes.

RESOLUTION

Dragon9 team has commented that gas estimation logic has been tested under multiple scenarios. The range of gas consumption is predictable, and there is no practical need for further adjustment. Inaccuracies in gas costs are considered highly unlikely given the current deployment conditions.

Identifier	Definition	Severity
LOG-02	Handling of non-resolvable addresses	Minor

If a provided wallet address is non-resolvable, smart contract returns empty address. This may cause issues in external systems or subsequent contract interactions if the empty address isn't handled correctly.

RECOMMENDATION

Ensure that any operations interacting with non-resolvable addresses fail gracefully or provide a clear error message.

RESOLUTION

Dragon9 team commented that non-resolvable addresses are rare edge cases and that the TON ecosystem handles `addr_none` as part of its normal address resolution protocol.

Identifier	Definition	Severity
LOG-03	Fallback to default on operation code mismanagement	Medium 🟡

Smart contract defaults to an exception throw if an unrecognized operation is received:

```
throw(0xffff);
```

This generic handling does not differentiate between various types of invalid operations, potentially leading to generic errors.

RECOMMENDATION

Use granular error handling and reporting for unrecognized or unsupported operations to assist in monitoring and quick response to attacks.

Identifier	Definition	Severity
LOG-05	Weak query ID handling	Minor

Query IDs are used in certain operations but are not rigorously validated. Weak or reused query IDs might leave the contract vulnerable to replay attacks.

`provide_wallet_address()`

RECOMMENDATION

Strengthen query ID validation to ensure they are unique and prevent potential replay attacks or message duplication.

RESOLUTION

Dragon9 team commented that risk of query ID reuse or replay attacks is negligible due to the asynchronous nature of the TON blockchain and the uniqueness of each interaction with the contract.

Identifier	Definition	Severity
LOG-06	Incomplete burn validation	Minor

Smart contract handles token burning through the following operation:

```
if (op == op::burn_notification()) {
    throw_unless(74,
        equal_slices(calculate_user_jetton_wallet_address(from_address, my_address(),
jetton_wallet_code), sender_address)
    );
    ...
}
```

Burn operation relies on correct address calculations and validations. A flaw in the address calculation as discussed in LOG-02 can lead to unauthorized or unintended burns.

RECOMMENDATION

Add logging to track burn events for auditability and trace any unexpected behavior, as discussed in COD-12.

Identifier	Definition	Severity
COD-01	Potential race conditions in `recv_internal()`	Minor

Multiple types of operations are processed inside the `recv_internal()` function. If external messages are processed out of order, this can introduce race conditions, particularly when minting or burning tokens.

RECOMMENDATION

Review message processing order and ensure that race conditions cannot arise from concurrent or asynchronous message handling.

RESOLUTION

Dragon9 team asserted that race conditions are not a concern due to the design of the TON virtual machine (TVM), which processes messages in a deterministic manner. The asynchronous nature of TON ensures that operations occur in sequence without the risk of race conditions.

Identifier	Definition	Severity
COD-10	Direct and indirect dependencies	Unknown

Smart contract interacts with external protocols and libraries, such as imported *FunC* libraries, third-party `jetton` wallet code, and the TON ecosystem for message handling and wallet interactions. For the purposes of this audit, these external dependencies are treated as black boxes, and their functional correctness is assumed. However, in the real-world environment, these external components could be compromised or exploited.

Additionally, upgrades or changes in these third-party entities—such as updates to wallet code or changes in message-handling protocols—could introduce severe impacts, including increased gas fees, incompatibility issues, or even failure in message forwarding.

RECOMMENDATION

Inspect third party dependencies regularly, and mitigate severe impacts whenever necessary.

ACKNOWLEDGEMENT

Dragon9 team will inspect third party dependencies regularly, and push upgrades whenever required.

Identifier	Definition	Severity
COD-12	Lack of event-driven architecture	Minor

Smart contract uses function calls to update state, which can make it difficult to track and analyze changes to the contract over time.

RECOMMENDATION

Use events to track state changes. Events improve transparency and provide a more granular view of contract activity.

ACKNOWLEDGEMENT

Dragon9 team has acknowledged this finding, and kept the code as-is.

DISCLAIMERS

InterFi Network provides the easy-to-understand audit of blockchain source codes (commonly known as smart contracts).

The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. This audit report could include false positives, false negatives, and other unpredictable results.

CONFIDENTIALITY

This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.

NO FINANCIAL ADVICE

This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way

to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

TECHNICAL DISCLAIMER

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, INTERFI NETWORK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT'S OR ANY OTHER INDIVIDUAL'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

TIMELINESS OF CONTENT

The content contained in this audit report is subject to change without any prior notice. InterFi Network does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.

LINKS TO OTHER WEBSITES

This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than InterFi Network. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites' and social accounts' owners. You agree that InterFi Network is not responsible for the content or operation of such websites and social accounts and that InterFi Network shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.

ABOUT INTERFI NETWORK

InterFi Network provides intelligent blockchain solutions. We provide smart contract development, testing, and auditing services. We have developed 150+ solidity codes, audited 1000+ smart contracts, and analyzed 500,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Velas, Oasis, etc.

InterFi Network is built by engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 4 core members, and 6+ casual contributors.

Website: https://interfi.network

Email: hello@interfi.network

GitHub: https://github.com/interfinetwork

Telegram (Engineering): https://t.me/interfiaudits

Telegram (Onboarding): https://t.me/interfisupport

interfinetwork

hello@interfi.network

https://interfi.network

SMART CONTRACT AUDITS | SOLIDITY DEVELOPMENT AND TESTING

RELENTLESSLY SECURING PUBLIC AND PRIVATE BLOCKCHAINS