SPYWOLF

Security Audit Report

Audit prepared for

HuntFi

Completed on

Aug 30, 2025

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

TABLE OF CONTENTS

PROJECT INFORMATION

- **Project name:** HuntFi (HUFI)
- **Website:** https://www.huntfi.app/
- **Network:** TON (The Open Network)
- **Purpose:** Location-based (“GPS treasure hunt”) game that rewards players with HUFI jettons. Ongoing **minting** is part of gameplay/emissions.

Smart-contract scope (in this audit)

- **Jetton Master (discoverable) – FunC**
  - File(s): jetton_minter_discoverable.fc, utils.fc, discovery-params.fc, imports/op-codes.fc, stdlib.fc
  - Responsibilities: minting, burn notifications, wallet-address discovery, admin & metadata management.
  - Upgradeability: **no set_code path** observed (master not upgradeable).

- **Jetton Wallet**
  - Implemented as a code cell stored in the master’s data at deploy time.
  - **Status: Verified**.

Out of scope: backend/game servers, website security, presale web flows, mobile apps, non-HUFI contracts, infra/ops.

On-chain artifacts

- **Jetton Master (friendly):** EQDAvU0KOeCg83BHWyiAGmgax4uI8u-OwHWI5BNxCo8WgmUe
- **Jetton Master (raw workchain:addr):** 0:c0bd4d0a39e0a0f370475b28801a681ac78b88f2ef8ec075a5e413710a8f1682
- **Admin address (friendly):** UQA_D0fKgWu608YIPDfsPf22IqBW7IlQUkWjaWeu862DBoGL
- **Jetton standard:** Jetton master with **discoverability** (provide/take wallet address ops).
- **Language / toolchain:** FunC (TON VM).

Token parameters (as provided)

- **Name / Symbol:** HuntFi / HUFI
- **Decimals:** 9
- **Supply model: Uncapped, emission-based** (ongoing minting tied to real-world gameplay; **no hard cap**)
- **Metadata policy: Mutable** to support seasonal events, city expansions, and live ops (visuals/content may change; core identity stays consistent).

SCOPE OF AUDIT(1)

Objectives

- Determine if the **HUFI Jetton** contracts are safe for a **presale** and **ongoing emissions**.
- Identify security/operational risks and propose mitigations.

In Scope

Smart contracts (FunC)

- **Jetton Master (discoverable)** — jetton_minter_discoverable.fc
  Address: EQDAvU0KOeCg83BHWyiAGmgax4uI8u-OwHWl5BNxCo8WgmUe
  Roles: admin-gated mint, burn notifications, wallet discovery, admin/metadata changes.
- **Libraries:** utils.fc, discovery-params.fc, imports/op-codes.fc, stdlib.fc.

Jetton Wallet

- Code cell referenced by master (immutable after deploy).
- **Status:** Verified by client against a known-good wallet (hash match asserted).

State & metadata

- Storage: total_supply, admin_address, content, jetton_wallet_code.
- Metadata JSON provided by client.

Out of Scope

- Website/app (**huntfi.app**), backend/game servers, emissions backend/controller, infra & key mgmt.
- Third-party services, bridges/CEX, KYC/AML, formal proofs, gas benchmarking.
- Any contracts not listed above.

Assumptions

- Wallet code hash equals a standard audited wallet; behavior is canonical.
- Deployed master matches provided code and is **not upgradeable**.
- Ongoing **minting is intended** and governed by the project.

Threat Coverage

- Access control (mint/admin/content), supply accounting (mint/burn), discovery flow.
- Message parsing/bounce handling, wallet address derivation, fee modes.
- Invariants and failure paths; admin/ownership risks; operational funding thresholds.

Deliverables

- **Findings & severity**, **recommendations**, **verification artifacts**.

AUDIT METHODOLOGY(1)

Approach

We performed a **targeted smart-contract audit** of the HUFI Jetton on TON with the goal of validating safety for a **presale** while supporting **ongoing emissions**. The review emphasizes correctness, access control, supply integrity, and operational safety rather than app/infra security.

Activities

1. **Specification alignment**
    - Collected project intent (minting as gameplay/emissions; discoverability required).
    - Mapped intended behavior to Jetton standard interfaces and ops.
2. **Manual code review (FunC)**
    - Read jetton_minter_discoverable.fc and support libs line-by-line.
    - Traced control flow for each op: **mint (21)**, **burn_notification**, **provide/take wallet address**, **admin change (3)**, **content change (4)**.
    - Checked storage layout & invariants: total_supply, admin_address, content, **immutable** jetton_wallet_code.
3. **State-transition & message analysis**
    - Verified message parsing, flags, and **bounce** handling.
    - Reviewed send_raw_message modes and fee assumptions.
    - Confirmed address derivation for wallets and burn-origin validation.
4. **Standards conformance**
    - Compared interfaces/ops against common Jetton implementations (discoverability ops, getters).
    - Ensured expected getters (get_jetton_data, get_wallet_address) semantics.
5. **Risk modeling**
    - Identified trust boundaries (admin, metadata mutability, emissions controller if used).
    - Considered failure modes: underfunded master, malformed calls, replay/abuse vectors in mint flow.
6. **On-chain artifact checks**
    - Cross-checked the provided **master address** and metadata.
    - Treated the **wallet code** as **verified by client** (hash matched to a known-good implementation).

AUDIT METHODOLOGY(2)

What we validated (checklist)

- **Access control**
    - Admin-only mint, admin change, and content change.
- **Supply accounting**
    - total_supply increases on admin mint; decreases on verified wallet burn.
- **Discoverability**
    - Requires sufficient attached TON; safe reply path and address derivation.
- **Upgradeability**
    - No code-upgrade hook in master; wallet code reference stored once and not mutated.
- **Error handling**
    - Reverts on unauthorized ops; ignores bounced messages; sensible fee checks.
- **Operational readiness**
    - Master must be funded for deploy/forward fees; UI must attach enough TON for discovery.

Severity model

- **Critical:** Loss of funds/control, unlimited supply without intended governance, or exploitable backdoor.
- **High:** Breaks core economics or user safety; likely to be hit under realistic conditions.
- **Medium:** Degrades reliability/accounting; mitigations or user education feasible.
- **Low/Info:** Minor inconsistencies, UX issues, or transparency/documentation gaps.

Each finding is rated by **Impact × Likelihood**, with context from intended emissions.

FINDINGS

🟥 Critical Risk

Centralized, Unlimited Mint Authority (Admin) ✅ Accepted by Design

Description: op::mint (21) allows the **admin** to mint any amount at any time. There is **no on-chain cap** or rate limit. This is by design for gameplay emissions, but concentrates absolute supply power in one key/contract.

Impact: A malicious/compromised admin (or buggy backend) can mint arbitrarily, collapsing price/liquidity and harming presale buyers.

Recommendation:: Make the master's **admin a dedicated EmissionsController** contract (not an EOA). Enforce signed tickets + per-epoch caps + rolling limits + per-tx ceilings, with **multisig ownership** and a **timelock** on parameter changes.

Single-Signer Admin Key Risk (If EOA) ✅ Mitigated

Description: If the current admin_address is a single EOA, loss/compromise enables unlimited minting and metadata changes.

Impact: Catastrophic, immediate loss of economic integrity.

Recommendation: Move admin to a **2+ multisig** (or the EmissionsController owned by a multisig). Use hardware keys, segregated roles, and rotate any exposed keys.

FINDINGS

🟥 High Risk

Off-Chain “Max Supply” Not Enforced On-Chain ✅ Resolved

Description: Metadata declares “Max.supply: 1,000,000,000 HUFI” but the master enforces **no cap**.

Impact: Users may rely on a cap that doesn’t exist; reputational and regulatory risk if supply exceeds 1B.

Recommendation: Update disclosures to **“uncapped, emission-based”** (or move to a controller-enforced cap if desired). Publish your emission policy and live dashboards for transparency.

Mutable Metadata by Admin (op = 4) ✅ Design Choice

Description: Admin can update content at any time (name/symbol/URI/image).

Impact: Potential user deception (e.g., swapping name/icon/URI), phishing via metadata, or listing confusion.

Recommendation: Host metadata on **IPFS/Arweave**; commit and publish content hash. Limit who can change it (via controller + timelock), and document policy. If feasible later, **renounce** admin when policy allows.

FINDINGS

🟥 High Risk

No On-Chain Emission Policy (Caps/Rate Limits) ✅ Design Choice

Description: The master has no notion of epoch/rolling caps, deadlines, or nonces for mints. All governance is off-chain trust.

Impact: Accidental or burst minting can outpace liquidity and game economics; hard to audit emissions.

Recommendation (short): Implement the **EmissionsController** with: signed mint tickets (to, amount, deadline, nonce), **nonce replay protection**, **per-epoch cap**, **24h rolling cap**, **pause()**, and clear events. Make the controller the **admin**.

No Native Pause/Emergency Stop in Master

Description: The master cannot be paused; if an exploit/abuse is found, minting continues unless admin is rotated or contract balance is drained.

Impact: Ongoing damage during incidents.

Recommendation (short): Put minting behind a **Controller with pause()** and a **guardian** role (pause-only). Publish controller address and paused state; monitor and alert on state changes.

FINDINGS

🟥 High Risk

Admin Rotation Without Safeguards (op = 3)

✅ Changes are pre-announced

Description: Admin can switch to any address immediately; absent a timelock/announce period, a hostile rotation can occur unnoticed.

Impact: Stealth takeover of mint/metadata powers.

Recommendation (short): Require admin changes via **multisig + timelock**, emit/log events, and **publicly pre-announce** rotations. Monitor and alert on op = 3 messages to the master.

FINDINGS

🟧 Medium Risk

Missing Supply Underflow Guard on Burn

Description: On op::burn_notification, the master subtracts jetton_amount from total_supply without explicitly checking total_supply >= jetton_amount. It relies on wallet correctness to avoid underflow.

Impact: If a malformed/forged burn passed wallet-checks due to unforeseen edge cases, supply accounting could corrupt.

Recommendation: Add throw_unless(76, total_supply >= jetton_amount); before subtraction.

Supply Increment Trusts Unvalidated master_msg Amount

Description: During mint, total_supply increases by the jetton_amount parsed from master_msg; there’s no assertion that this matches the intended amount or expected opcode/structure.

Impact: An incorrectly constructed master_msg (even by mistake) could mis-account supply vs. user-visible “minted” amounts.

Recommendation: Sanity-check master_msg (expected op code/layout) and, if applicable, assert intended amount equality. Consider eliminating duplicated amount fields.

FINDINGS

🟧 Medium Risk

Hardcoded Discovery Gas Threshold (≈0.01 TON)

Description: provide_wallet_address requires msg_value > fwd_fee + 0.01 TON based on a static estimate. Network cost drift can cause valid queries to fail or waste excess.

Impact: UX friction / intermittent failures for wallet discovery.

Recommendation: Make the threshold configurable (via controller-admin) or adopt a safer margin and document the required attached TON in the dApp.

No Min-Value Check for op::mint Forwarding

Description: The master forwards TON (amount) to the recipient wallet but does not verify sufficiency for deploy/forward costs.

Impact: Underfunded mints can fail/bounce, disrupting emissions/presale flows.

Recommendation: Enforce a minimum amount or validate msg_value/master balance before sending; document required TON in off-chain mint flows.

FINDINGS

🟧 Medium Risk

Operational Dependence on Master Balance

Description: mint_tokens (mode 1) and discovery replies consume the master's TON. If depleted, mints/responses fail.

Impact: Emissions/presale interruptions and user support load.

Recommendation: Set up balance monitoring + auto-top-up; alert on low-balance thresholds.

No Pause/Emergency Stop at Master Level

Description: Master lacks a native pause. (Minting can be indirectly halted only by changing admin or draining funds.)

Impact: Incident response is slower; ongoing harm during exploits/misconfigurations.

Recommendation: Put mint behind a **Controller** with pause() and guardian key; make Controller the admin.

FINDINGS

🟨 Low Risk

Limited Validation of master_msg Opcode

Description: The code assumes master_msg's structure (skips 32+64 then reads amount) but doesn't verify the opcode matches the expected wallet entrypoint.

Impact: If a malformed payload is used, unexpected wallet behavior or supply accounting mismatch can occur before revert.

Recommendation: Check the opcode in master_msg against the intended wallet call pattern.

Observability & Transparency Gaps

Description: No built-in telemetry/events beyond standard messages; emissions transparency is purely off-chain.

Impact: Harder community auditing and slower incident detection.

Recommendation: Publish a live dashboard (total supply, 24h minted, controller state), and alert on op=21/3/4 messages.

FINDINGS

🟨 Low Risk

Discovery Flow UX Sensitivity

Description: Users/dApps must attach enough TON for provide_wallet_address; shortfalls cause rejections.

Impact: Support burden and failed UI actions.

Recommendation: In the dApp, prefill a safe TON amount and explain fees; retry with higher value on failure.

CONCLUSION

Verdict: Approved for presale (design choices acknowledged).
HUFI's Jetton master is a standard, non-upgradeable implementation with a **verified wallet**. The project **intentionally** runs an **uncapped, emission-based** model and **mutable metadata** to support seasonal events and content updates. Emissions are **adaptive to real-world activity** and backed by backend **rate limits/anti-bot** measures. **Admin changes are rare and pre-announced.** These are product decisions, not contract flaws; the primary risks are **governance and operations**, not bugs.

What changed from the draft

- "Max supply = 1B" removed; supply is **uncapped by design**.
- Metadata mutability reclassified as a **game feature** (seasonal/city expansions), not a vulnerability.
- Emission policy recognized as **dynamic**; emphasis shifts to **disclosure and monitoring** rather than on-chain caps.
- Admin rotation risk **downgraded** given the team's practice of pre-announcements.

Conditions for a safe launch

1. **Governance & key custody**
   - Prefer a **multisig-owned controller** (or multisig admin). If a short-term EOA is unavoidable, use hardware custody and 24/7 monitoring; migrate to multisig promptly.
2. **Disclosure**
   - Update metadata/website to state **"uncapped, emission-based"**. Publish master address, master code hash, **wallet code hash**, and (if used) controller address/policy.
3. **Transparency & monitoring**
   - Provide a live dashboard (**total supply**, **24h minted**). Alert on **mint (op=21)**, **admin change (op=3)**, **content change (op=4)**, controller parameter changes, and **low TON balance**.
4. **Operational readiness**
   - Keep the master adequately **funded** for wallet deploy/forwards. Ensure the dApp attaches **slightly >0.01 TON** for discovery and handles retries/clear errors. Maintain an incident runbook; if using a controller, include a **pause/guardian**.

Optional hardening (non-blocking)

- Add an explicit **underflow guard** on burn and **opcode/shape validation** for master_msg.
- Pin key assets/metadata to **IPFS/Arweave** and keep a public **changelog** for major visual/URI updates.

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ OVER 700 SUCCESSFUL CLIENTS
- ✓ MORE THAN 1000 SCAMS EXPOSED
- ✓ MILLIONS SAVED IN POTENTIAL FRAUD
- ✓ PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS
- ✓ CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.

While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.