verichains

SECURITY AUDIT OF

FUNARCADE TOKEN SMART CONTRACT

Public Report

Jan 09, 2024

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Jan 09, 2024. We would like to thank the Funarcade Token for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Funarcade Token Smart Contract. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issue in the contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Funarcade Token Smart Contract

FAT is a ERC-20 token on the Arbitrum blockchain. FAT can be staked to earn a portion of the platform profits or used for play in their exciting games.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Funarcade Token Smart Contract. It was conducted on commit `b19c4a5a023bc87d83841483dd26fef774251c95` from git repository *https://github.com/FunAsiaSolution/FunAsia-Crypto-Token*.

The latest version of the following file was made available during the review process:

SHA256 Sum	File
`3e52ed18e0aad9b690db9719f6efea0a13172f21982d48465253fa83c83be1e8`	`FAT.sol`

1.3. Audit methodology

Our security audit process includes four steps:

- Mechanism Design is reviewed to look for any potential problems.
- Source codes are scanned/tested for commonly known and more specific vulnerabilities using public and our in-house security analysis tool.
- Manual audit of the codes for security issues. The source code is manually analyzed to look for any potential problems.
- Set up a testing environment to debug/analyze found issues and verifies our attack PoCs.

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the functioning; creates a critical risk to the application; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the application with high impact; needs to be fixed with high priority.

SEVERITY LEVEL	DESCRIPTION
MEDIUM	A vulnerability that could affect the desired outcome of executing the application with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

Funarcade Token acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. Funarcade Token understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, Funarcade Token agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the Funarcade Token will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the Funarcade Token, the final report will be considered fully accepted by the Funarcade Token without the signature.

2. AUDIT RESULT

2.1. Overview

The Funarcade Token Smart Contract was written in `Solidity` language, with the required version to be `^0.8.9`.

2.1.1. FAT.sol

The `FAT` token contract extends `ERC20` and `Ownable` contracts. With `Ownable`, by default, the contract owner is the contract deployer, but he can transfer ownership to another address at any time.

The address `_checker` is capable of transferring funds from any user as long as the owner has not turned off `_isChecker`.

Since we do not control the logic of the `_checker` contract, there is no guarantee that `_checker` will not contain any security related issues.

Table 2 lists some properties of the audited Token Smart Contract (as of the report writing time).

PROPERTY	VALUE
Name	Funarcade Token
Symbol	FAT
Decimals	18
Total Supply	100,000,000 (x10^{18}) Note: the number of decimals is 18, so the total representation token will be 100,000,000 or 100 million.

Table 2. The Token Smart Contract properties

For the `ERC20` token, the security audit team has the list of centralization issues below:

Checklist	Status	Passed
Upgradeable	No	Yes
Fee modifiable	Yes	

Checklist	Status	Passed
Mintable	No	Yes
Burnable	No	Yes
Pausable	No	Yes
Trading cooldown	No	Yes
Has blacklist	Yes	
Has whitelist	No	Yes

Table 3. The decentralization checklist

2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of Funarcade Token Smart Contract.

2.3. Additional notes and recommendations

2.3.1. Redundancy of variables and events, and the absence of necessary events

- The variable `deadAddress` is initialized but not used.
- The event `UpdateUniswapV2Router()` is defined but not used.
- State-changing functions of the contract, such as update functions, `enableTrading()`, `removeLimits()`, `excludeFromMaxTransaction()`, `withdrawStuckToken()`, `withdrawStuckEth()`, `renounceBlacklist()`, `blacklist()`, `blacklistLiquidityPool()`, `unblacklist()`, `turnOffChecker()`, lack the necessary events.

APPENDIX

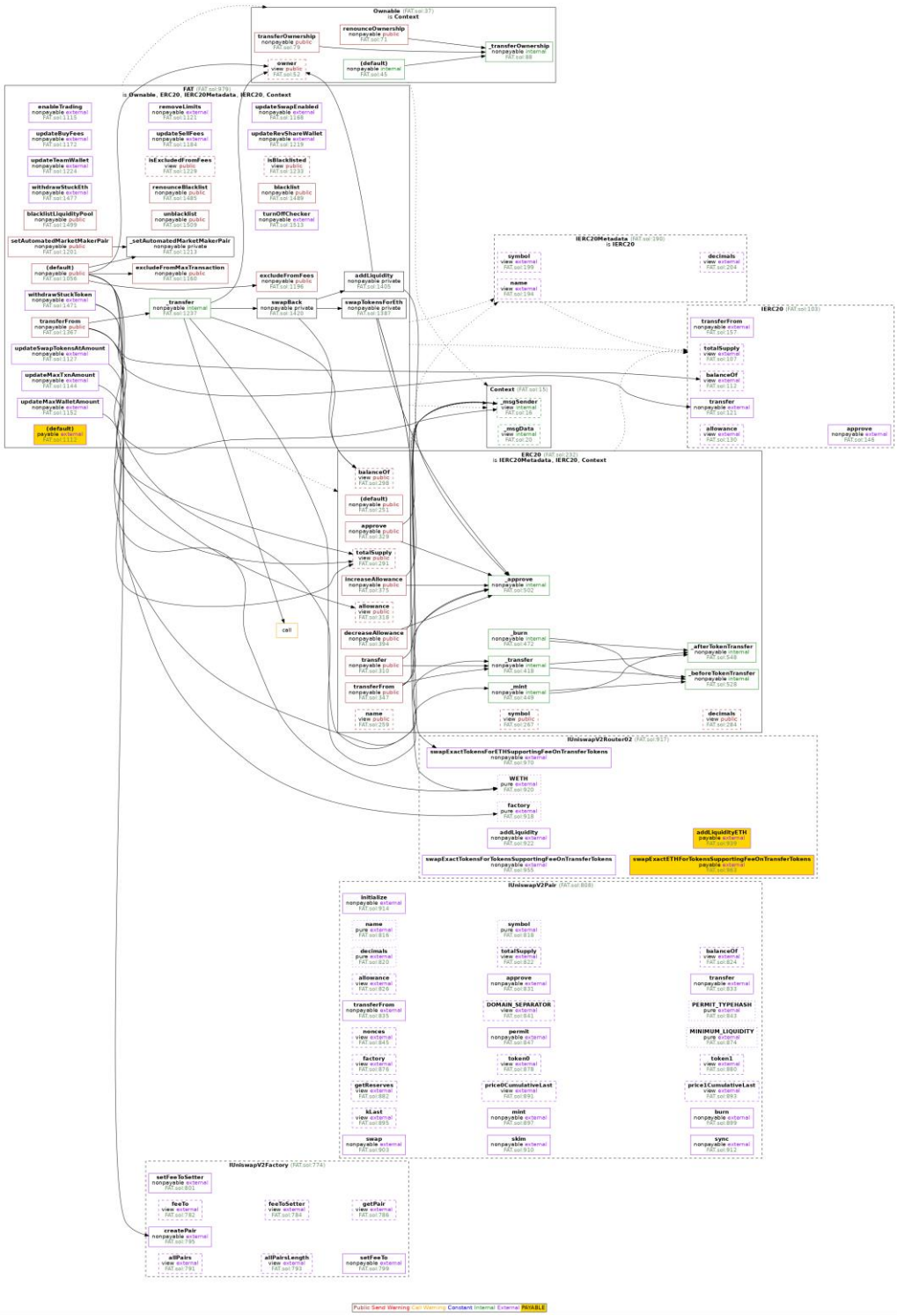

Image 1. Funarcade Token Smart Contract call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Jan 09, 2024*	Public Report	Verichains Lab

Table 4. Report versions history