verichains

SECURITY AUDIT OF

FADOGO TOKEN SMART CONTRACTS

Public Report

Feb 08, 2022

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 prepared by Verichains Lab on Feb 08, 2022. We would like to thank the FadoGo 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 FadoGo Token Smart Contracts. 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 issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About FadoGo Token Smart Contracts

FADO Go is setting the next generation of livestream e-commerce and cryptocurrency.

By taking a hybrid approach that combines the best practices of cross-border e-commerce platforms with social emerging technologies, FADO Go is built with a modern social technology-based international trade method that directly connects manufacturers with consumers and makes livestream cross-border shopping as simple as buying on social media platforms. We utilize blockchain to take our service quality to a whole new level with new decentralized features and differentiate ourselves apart from the competition of global e-commerce.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of FadoGo Token Smart Contracts. It was conducted on commit 5622eac989241518d2bc3f458306328db551d4d3 from git repository *https://github.com/FadoGo/FADO-Token*.

There are 2 files in our audit scope. They are FadoToken.sol, FgtToken.sol files.

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference

- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

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 contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract 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

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The following files were made available in the course of the review: FadoToken.sol, FgtToken.sol.

2.2. Contract code

The FadoGo Token Smart Contracts was written in Solidity language, with the required version to be ^0.8.2. The source code was written based on OpenZeppelin's library.

2.2.1. FadoToken contract

FadoToken contract is an ERC20 token contract. The contract extends AccessControlEnumerable, ERC20Pausable, ERC20Capped and ERC20Burnable contracts. With AccessControlEnumerable, by default, the contract deployer has DEFAULT_ADMIN_ROLE, MINTER_ROLE and PAUSER_ROLE roles. The user has PAUSER_ROLE can pause/unpause contract using ERC20Pausable contract, user can only transfer tokens when the contract is not paused. ERC20Burnable allows token holders to destroy both their own tokens and those that they have an allowance for.

The contract also implements some functions which call to a BPContract to control the transfer action. The mint function allows the user has MINTER_ROLE to create new tokens which change the totalSupply of the contract in the maxSupply range.

2.2.2. FgtToken contract

FgtToken contract is an ERC20 token contract. The contract extends AccessControlEnumerable, ERC20Pausable, ERC20Capped and ERC20Burnable contracts. With AccessControlEnumerable, by default, the contract deployer has DEFAULT_ADMIN_ROLE, MINTER_ROLE and PAUSER_ROLE roles. The user has PAUSER_ROLE can pause/unpause contract using ERC20Pausable contract, user can only transfer tokens when the contract is not paused. ERC20Burnable allows token holders to destroy both their own tokens and those that they have an allowance for.

The contract also implements some functions which call to a BPContract to control the transfer action. The mint function allows the user has MINTER_ROLE to create new tokens which change the totalSupply of the contract in the maxSupply range.

2.3. Findings

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

2.4. Additional notes and recommendations

2.4.1. BPContract function INFORMATIVE

Both two contracts in the audit scope use BPContract to control the transfer action. Since we do not control the logic of the BPContract, there is no guarantee that BPContract will not contain any security related issues. With the current context, in case the BPContract is compromised, there is not yet a way to exploit the FadoGo Token Smart Contracts, but we still note that here as a warning for avoiding any related issue in the future.

By the way, if having any issue, the BPContract function can be easily disabled anytime by the DEFAULT_ADMIN_ROLE user using the setBpEnabled function. In addition, BPContract is only used in a short time in token public sale IDO then the DEFAULT_ADMIN_ROLE user will disable it forever by the setBPDisableForever function.

2.4.2. Token can be minted by the public function INFORMATIVE

Both two contracts in the audit scope implement the mint public function that allows the MINTER_ROLE to mint new tokens. The totalSupply of the contract can be changed by this function. Although, the contract inherits the ERC20Capped to set maxSupply - the totalSupply limitation. The contract may still get dangerous if the private key of the MINTER_ROLE user is lost.

RECOMMENDATION

We recommend pre-minting tokens in the constructor and removing the mint public function.

APPENDIX

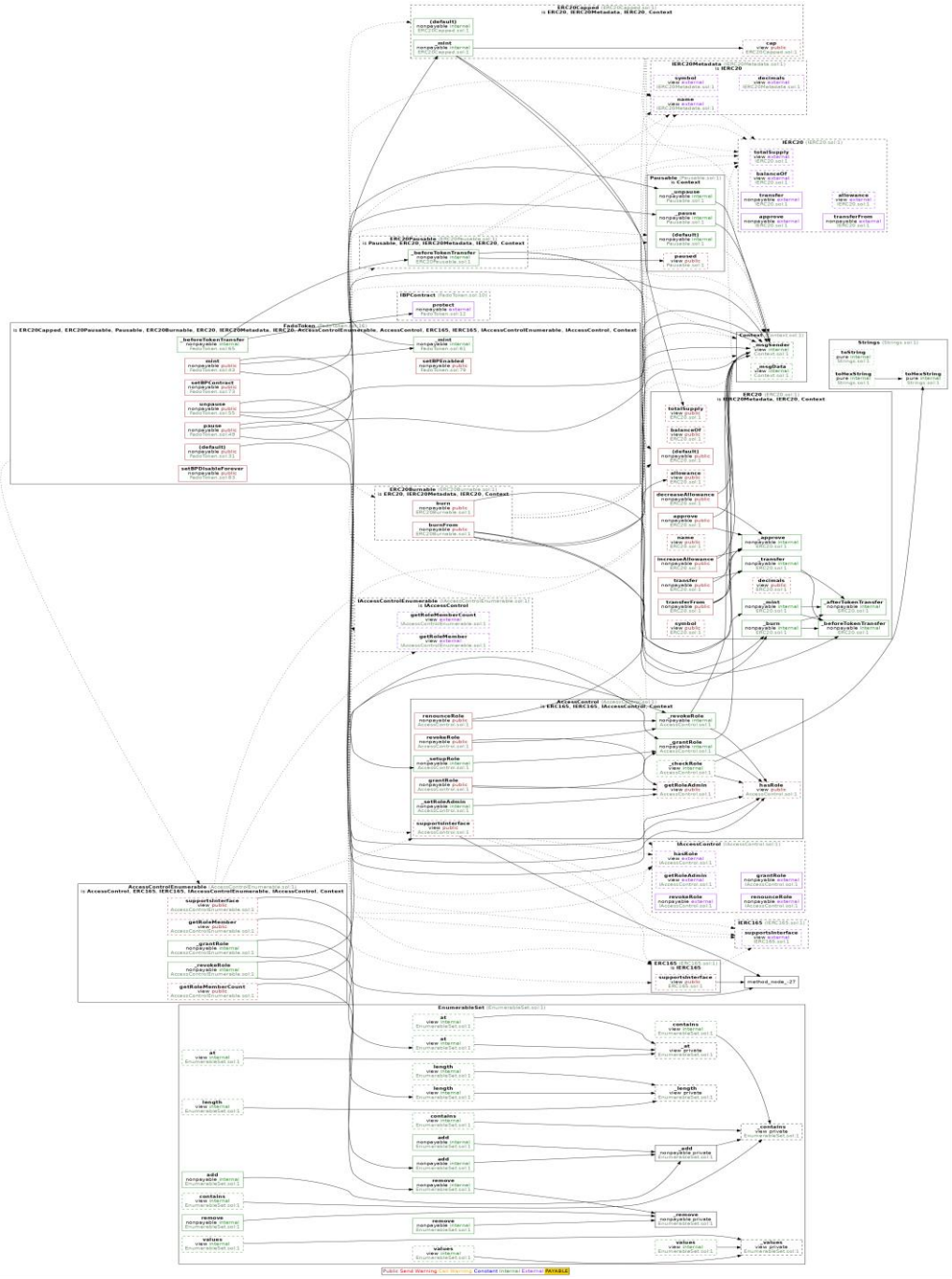

Image 1. FadoToken Smart contracts call graph

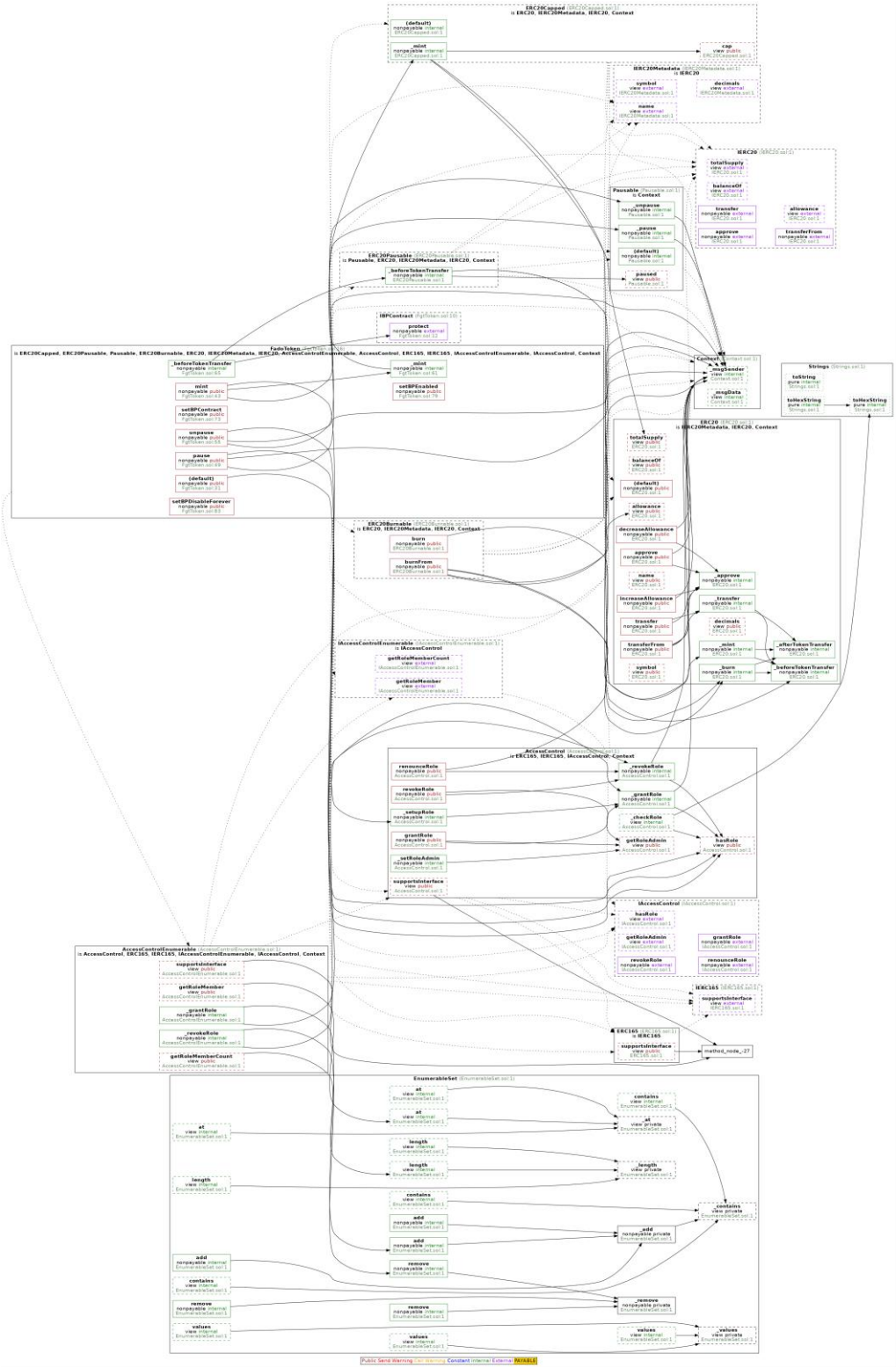

Image 2. FgtToken Smart contracts call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Feb 08,2022*	Public Report	Verichains Lab

Table 2. Report versions history