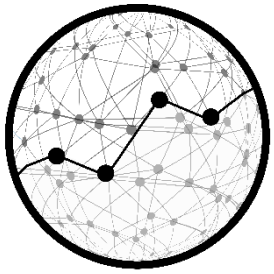

ICORATING
THE RATING AGENCY FOR
THE ICO MARKET

Smart Contract Audit of Fort Financial Crypto

Contact:

Mikhail Mironov, Head of Partnerships at ICORating, Amsterdam, +31 6 83624781, mikhail@icorating.com

TABLE OF CONTENTS

Contact:
Mikhail Mironov, Head of Partnerships at ICORating, Amsterdam, +31 6 83624781, mikhail@icorating.com

1. DISCLAMER

The information contained in this document is for informational purposes only. The views expressed within are solely the personal stance of the ICORating team, based on data available in open access and information that the developers provided to the team through Skype, email or other means of communication. Our goal is to increase the transparency and reliability of the young ICO market and to minimize the risk of fraud. We appreciate feedback with constructive comments, suggestions, and ideas on how to make our analysis more comprehensive and informative.

Contact:
Mikhail Mironov, Head of Partnerships at ICORating, Amsterdam, +31 6 83624781, mikhail@icorating.com

2. INTRODUCTION

2.1 Vulnerability Level For Technical Audit

- **Critical severity** – A vulnerability that could disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.
- **High severity** – A vulnerability that affects the desired outcome when using a contract, or provides an opportunity to use the contract in unintended ways.
- **Medium severity** – A vulnerability that could affect the desired outcome of executing the contract in certain scenarios.
- **Low severity** – A vulnerability that does not have a significant impact on the use of the contract and is probably subjective.

3. MAIN RESULTS

3.1 General comments

The code which is being audited is located in the FortFC/Contract repository. The commit used to audit is 44813b651caf08d27b6e036d90f6f1a6dd50459e.

The purpose of the audit was to:

– ensure that the smart contract is functioning correctly.

– identify potential security issues.

The information contained in this report should be used to understand the impact of risks on a smart contract and as a guide to improving the security level of a smart contract by eliminating the problems identified.

Contact:
Mikhail Mironov, Head of Partnerships at ICORating, Amsterdam, +31 6 83624781, mikhail@icorating.com

3.1.1 Critical severity

✕ None	✓ None

3.1.2 High severity

✕ None	✓ None

3.1.3 Medium severity

✕ None	✓ None

3.1.4 Low severity

✕ The version of Solidity used in the smart contracts is out-of-date.	✓ Use the latest version of Solidity, because the developers have fixed bugs that may be relevant to the version that your smart contracts use.
✕ Variable ICOEnd is set for 2024, however on the website the date is 2019. In the smart contract uploaded to the network, the date has been set correctly as it is on the website.	
✕ Variable TransferEnable does not have any practical value; the contract's logic can exist without this variable.	✓ This variable can be deleted and its application removed from **modifier transferEnable**.
✕ Known vulnerabilities of ERC-20 token.	✓ 1. It is possible to perform a double withdrawal attack. More details here. ✓ 2. Lack of transaction handling mechanism issue. More details here.

4. CONCLUSION

4.1 Technical conclusion

The smart contract has a several low severity issues. These issues do not affect the functionality of the smart contract.

The smart contract is free of critical issues and ready for publication on the mainnet.

Contact:
Mikhail Mironov, Head of Partnerships at ICORating, Amsterdam, +31 6 83624781, mikhail@icorating.com

5. TOOLS USED

The audit of the contract code was conducted using Remix IDE (http://remix.ethereum.org).

Securify (https://securify.ch) and SmartCheck (https://tool.smartdec.net) services were also used to verify the smart contract for known errors.

The Solidity compiler version used was 0.4.24 (the most recent stable version).