CERTIK

Security Assessment

Forbitspace - Dex aggregator

Nov 19th, 2021

Table of Contents

Summary

This report has been prepared for Forbitspace Foundation LLC to discover issues and vulnerabilities in the source code of the Forbitspace - Dex aggregator project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Forbitspace - Dex aggregator
Platform	Ethereum
Language	Solidity
Codebase	Forbitspace Foundation LLC Company
Commit	https://github.com/forbitspace/forbitspaceX-erc20/tree/12350701eb1653b72b748527635e56d623f20e64 https://etherscan.io/address/0xb3FeF4B71A4EDB6f1BD51bf9417876042B936dd6#code

Audit Summary

Delivery Date	Nov 19, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

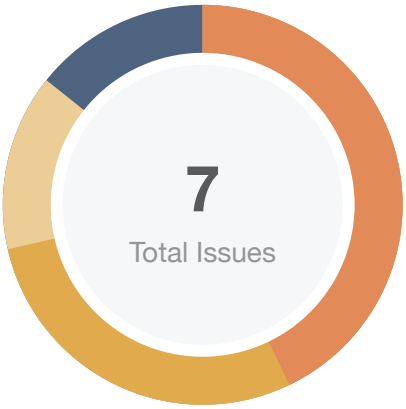
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	3	0	0	0	3	0
Medium	2	0	0	0	0	2
Minor	1	0	0	1	0	0
Informational	1	0	0	0	0	1
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
IER	contracts/interfaces/IERC20.sol	82e91992e7361e4cdf51b56d53e60621853d0c2b6d0d292be4e423a06b7745f9
IPX	contracts/interfaces/IPayment.sol	7a8a5889c8a8ffc63b07edd995ecb2d5ae3954837cca57ee6ca669b16660c444
IWE	contracts/interfaces/IWETH.sol	927bd26699150b8c1616ca5faf607707b41e5516ac1e82b71374000515e15f0e
IXX	contracts/interfaces/IforbitspaceX.sol	d583f1f1f0679882c353ac2857ac19f1100a4e4223c8f366a1ffb4cf2d9a8693
AXE	contracts/libraries/Address.sol	6261eae0bb66d120b457f5f13ddc008243ed3841a5a44444a90a61889932413b
CXE	contracts/libraries/Context.sol	26fff06a64358afb3b1408969acb21eb15490e124cefd718dbfd90ed1b4ae40b
OXE	contracts/libraries/Ownable.sol	28b1dc364e9d8ff79eb4b01bb737ce4be4cedfe36643a52ccf9a79f973d6f34d
PXE	contracts/libraries/Payment.sol	c811c36576950ce806cb24cb4cc2f185e8e60df1ea0c59fb2b4d50e9d16d5a12
SER	contracts/libraries/SafeERC20.sol	d81c04c658c2822a8745ef99bdc336576bd7d951517ba7a9e471b46a124fcdb8
SMX	contracts/libraries/SafeMath.sol	e775dcb3acced520f8841ea7df4c29c101dc22407d71988bdcdfffbb6e12602f
XXE	contracts/forbitspaceX.sol	6c34b08546872064a501ae4a6149da59893a81ac4b7eff3db890e55d37b11572

Findings



Critical	**0** (0.00%)
Major	**3** (42.86%)
Medium	**2** (28.57%)
Minor	**1** (14.29%)
Informational	**1** (14.29%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
PXE-01	Centralization Risk	**Centralization / Privilege**	**Major**	Partially Resolved
XXE-01	Approval of unverified contract	Volatile Code	Major	Partially Resolved
XXE-02	Third Party Dependencies	Volatile Code	Minor	Acknowledged
XXE-03	Potential Reentrancy Attack	Logical Issue	Medium	Resolved
XXE-04	Centralization Risk	**Centralization / Privilege**	**Major**	Partially Resolved
XXE-05	Redundant Code	Gas Optimization	Informational	Resolved
XXE-06	Incompatibility With Deflationary Tokens	Volatile Code	Medium	Resolved

PXE-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/libraries/Payment.sol: 66~81	Partially Resolved

Description

In the contract `forbitspaceX`, the role `owner` has the authority over the following function:

- collectETH
- collectTokens
- aggregate (calls `collectTokens` once the fee-deducted amount has been sent).

Any compromise to the `owner` account may allow the hacker to take all the funds collected in the contract from fees.

Recommendation

We advise the client to carefully manage the `owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

[Forbitspace Foundation LLC]:

We recognize a potential problem in using an address as the owner; we will transfer ownership of the owner to the multisig contract and under the control of the timelock contract.

XXE-01 | Approval of unverified contract

Category	Severity	Location	Status
Volatile Code	Major	contracts/forbitspaceX.sol: 14~20, 42, 70	Partially Resolved

Description

User can approve any contract with an allowance of `type(uint).max` by calling the `aggregate` function and calling a function in `params.exchangeTarget` that swaps `tokenIn` for `tokenOut`. However, this can only be taken advantage of if there are tokens remaining in the contract after all `aggregate`, `collectETH` and `collectTokens` calls.

Recommendation

Creating a whitelist of trusted DEXes that can be called for swapping will alleviate the issue.

Alleviation

[Forbitspace Foundation LLC]:

We have found the threat you mentioned, however, it is not possible for the user to approve any contract, in the interface we have configured for approval only through multicall and the approve type(uint) .max only allows the forbitspaceX contract itself with Dexs, but the forbitspaceX contract itself only has multi swap functionality and does not contain any funds. So we see that approving type(uint).max will not be a potential threat. Building a whitelist is very appropriate, but at this stage of development we may not be able to complete it in time and will be completed in the following updates.

We have changed approve amount token enough to swap instead of approve type(uint).max amount token.

XXE-02 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	Minor	contracts/forbitspaceX.sol: 75	Acknowledged

Description

The contract is serving as the underlying entity to interact with DEX protocols and wETH. The scope of the audit treats 3rd party entities as black boxes and assume their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties can possibly create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Recommendation

We understand that the business logic of Payment requires interaction with DEXes and wETH. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

[Forbitspace Foundation LLC]:

We really appreciate your sincere advice and we understand the dependency when aggregating Dexs, we will continuously monitor Dexs and update them as quickly as possible.

XXE-03 | Potential Reentrancy Attack

Category	Severity	Location	Status
Logical Issue	Medium	contracts/forbitspaceX.sol: 1~84	Resolved

Description

A reentrancy attack can occur when the contract creates a function that makes an external call to another untrusted contract before resolving any effects. If the attacker can control the untrusted contract, they can make a recursive call back to the original function, repeating interactions that would have otherwise not run after the external call resolved the effects.

Recommendation

We recommend using the Checks-Effects-Interactions Pattern to avoid the risk of calling unknown contracts or applying OpenZeppelin ReentrancyGuard library - `nonReentrant` modifier for the aforementioned functions to prevent reentrancy attack.

Alleviation

[Forbitspace Foundation LLC]:

We used ReentrancyGuard library and checked the data in function aggregate as you mentioned.

XXE-04 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	contracts/forbitspaceX.sol: 52~56	Partially Resolved

Description

In the contract `forbitspaceX`, the role `owner` has the authority over the following function:

- collectETH
- collectTokens
- aggregate (calls `collectTokens` once the fee-deducted amount has been sent).

Any compromise to the `owner` account may allow the hacker to take all the funds collected in the contract from fees.

Recommendation

We advise the client to carefully manage the `owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

[Forbitspace Foundation LLC]:

We recognize a potential problem in using an address as the owner; we will transfer ownership of the owner to the multisig contract and under the control of the timelock contract.

XXE-05 | Redundant Code

Category	Severity	Location	Status
Gas Optimization	Informational	contracts/forbitspaceX.sol: 77~80	Resolved

Description

Can directly compare the current and previous balances in the require statement.

Recommendation

In the require statement, use inequalities given by `balanceOf(tokenX)` and previous balances `amountXActual` in the require statement rather than subtracting the values and checking if they're greater than 0.

Alleviation

[Forbitspace Foundation LLC]:

We have removed the redundant elements.

XXE-06 | Incompatibility With Deflationary Tokens

Category	Severity	Location	Status
Volatile Code	Medium	contracts/forbitspaceX.sol	Resolved

Description

There will be issues when `tokenIn` is standard ERC20 deflationary tokens. For example, let's assume `amountInTotal` is 10 and the token has a 10% tx fee. then `amountInActual` would be 9. If all 9 tokens are swapped, then in line 45, `amountInActual` is still 9. In line 51, the contract is supposed to send 10-9=1 `tokenIn` to the msg.sender, but the contract does not have any tokens at the moment. therefore, the function call `aggregate` would always fail.

Recommendation

We recommend not using `amountInTotal` to check if there are remaining tokens which needs to be sent back to msg.sender.

Alleviation

[Forbitspace Foundation LLC]:

To clarify the problem we are facing, we will show the contract's operation flow as follows:

The operation flow of forbitspace aggregator: First, the swap data of Dexs such as uniswapv2, uniswapv3, sushiwap, curve, ... will be built at the (FE) interface and then be synthesized through the 'aggregate' method of the forbitspaceX contract. For exchanges with the same swap method as uniswapv2, uniswapv3, we will use the “swapTokenForExtactToken” method. Therefore, the amount of token in used will be the amount in Max. This amount is just approximate, not accurate because it depends on the possibility of price slippage. Next, when the forbitspaceX contract receives the data built at FE, forbitspaceX will not take the tx fee of the TokenIn passed in but will use 100% of the amount in total in the received data to swap on the Dexs. For the Dexs that have a swap method that is similar to uniswap v2 and v3, the amount of Token In to swap may be less than expected due to slippage when the transaction is pending. Consequently, after the swap is completed on each Dex, the amount in actuality may have a small balance that will be refunded for traders.

Solution for the problem posed by Audit Report: based on the example in the report, when forbitspaceX receive an amount in total of 10, we will use all 10 to swap instead of deducting 10% tx fee in the example. Next, in case of swap all on UniswapV2 swap, the data swap will be 10 tokens in and we will get 8

tokensOut. However, during the pending transaction, the slippage occurs and changes the ratio at the moment that we just need 9.5 tokens in to receive 8 taken out. Hence, the amount in actual for a swap at this time is only 9.5 and there will be a surplus of 10 - 9.5 = 0.5. At the moment, the line 45 amountInActual will be at 0.5 and it will be refunded to the trader, the transaction will work normally instead of “function call aggregate would always fail.”

fixed in commit 118f7de542563f720427b2a569a676cca599fd8a.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. 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. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR 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, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'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. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

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

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.