Security Assessment

DODO LimitOrder

Dec 15th, 2021

Table of Contents

Summary

This report has been prepared for DODO LimitOrder to discover issues and vulnerabilities in the source code of the DODO LimitOrder 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.

Additionally, this audit is based on a premise that all external smart contracts are safely implemented. And the following `sol` files are not within the scope of the audit:

- `intf/IERC20.sol`
- `lib/SafeMath.sol`
- `lib/SafeERC20.sol`
- `external/draft-EIP712.sol`
- `external/ECDSA.sol`
- `intf/IDODOApproveProxy.sol`
- `lib/InitializableOwnable.sol`

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	DODO LimitOrder
Platform	Ethereum, BSC, Polygon, Arbitrum, MoonRiver, Boba, Aurora
Language	Solidity
Codebase	https://github.com/DODOEX/dodo-limit-order/tree/main/contracts
Commit	520dccc1a0729eb73d6a8a593fada4ee92dac623 a94248a89daf554cc36b476c57e3b03050f538c6

Audit Summary

Delivery Date	Dec 15, 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	1	0	0	1	0	0
Medium	1	0	0	0	0	1
Minor	0	0	0	0	0	0
Informational	2	0	0	2	0	0
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
ADD	lib/ArgumentsDecoder.sol	ff0b368410f34e98e540318c63eade7e8e3b4bdfbb29574b616523c29a02d677
DOD	DODOLimitOrder.sol	ffed5c69b1676a31eaacc9d5ff4bddda3a792da486943cda5f85a2f42238f4a1
DOO	DODOLimitOrderBot.sol	27333107ffdee186a59fbbc80f85dfe557ccf11607abe5fdd2a654ff22a0c977

Understandings

Overview

DODOLimitOrder is a project that provides a service to fill the limit order created by users or market makers.

There are two order types, limit order created by the user and RFQ created by the market maker. Along with a single order, there are two roles, maker and taker. Maker is who creates an order with a specific price. The taker is who fills the above order.

The `ECDSA` algorithm is used to verify the order and the order can not be filled if it is expired.

Limit Order

The user creates an order with a specific price. The taker, who is in the white list, used the received maker tokens to swap out taker token by calling `DODORouteProxy` contract. The amount of the swapped out taker tokens must equal to or greater than the number of taker tokens which is actually needed in the transaction for filling limit order. The surplus of the swapped-out tokens, which is a service fee, is transferred to a specific `_TOKEN_RECEIVER_`. The limit order may or may not be filled at one time, so the order is filled by multiple takers. Filling limit order is free of fee.

RFQ

The market maker creates RFQ orders in bulk. The RFQ orders can be filled by the user or by the platform. The maker fee, maker tokens, and taker tokens are transferred to recipients when the taker fill amount is less than the taker amount in the order. The maker fee is charged by the `_FEE_RECEIVER_`.

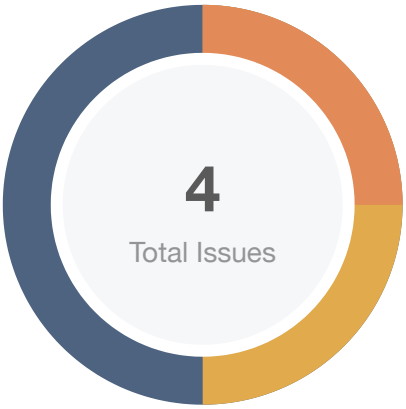
Privileged Functions

The contract contains the following privileged functions that are restricted by the `onlyOwner` modifier. They are used to modify the contract configurations and address attributes. We grouped these functions below:

The `onlyOwner` modifier:

- `addWhiteList()` in `DODOLimitOrder.sol`
- `removeWhiteList` in `DODOLimitOrder.sol`
- `changeFeeReceiver` in `DODOLimitOrder.sol`
- `addAdminList()` in `DODOLimitOrderBot.sol`
- `removeAdminList()` in `DODOLimitOrderBot.sol`
- `changeTokenReceiver()` in `DODOLimitOrderBot.sol`

Findings



Critical	**0** (0.00%)
Major	**1** (25.00%)
Medium	**1** (25.00%)
Minor	**0** (0.00%)
Informational	**2** (50.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
DOD-01	Missing Local Variables	Logical Issue	Medium	Resolved
DOD-02	Missing Input Validation	Volatile Code	Informational	Acknowledged
DOO-01	Missing Input Validation	Volatile Code	Informational	Acknowledged

GLOBAL-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Acknowledged

Description

The role `owner` has the authority over the listed functions:

- `addWhiteList()` in `DODOLimitOrder.sol`
- `removeWhiteList` in `DODOLimitOrder.sol`
- `changeFeeReceiver` in `DODOLimitOrder.sol`
- `addAdminList()` in `DODOLimitOrderBot.sol`
- `removeAdminList()` in `DODOLimitOrderBot.sol`
- `changeTokenReceiver()` in `DODOLimitOrderBot.sol`

Any compromise to the key role account may allow a potential hacker to take advantage of this and execute malicious acts.

Recommendation

We advise the client to carefully manage the key role 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 are some feasible suggestions that would also mitigate the potential risk at the different levels in terms of short-term and long-term scenarios:

- 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

No alleviation.

DOD-01 | Missing Local Variables

Category	Severity	Location	Status
Logical Issue	Medium	projects/DODO%20LimitOrder/contracts/DODOLimitOrder.sol (12a08a0): 129	Resolved

Description

The two `uint256` type return values of the linked function call should be separately assigned to the local variables `curTakerFillAmount` and `curMakerFillAmount`. Otherwise, both of the two local variables are always zero.

Recommendation

We advise refactoring the linked statement as below:

```
129 (curTakerFillAmount, curMakerFillAmount) =
_settleRFQ(order,filledTakerAmount,takerFillAmount,thresholdMakerAmount,taker);
```

Alleviation

The development team solved this issue at commit a94248a89daf554cc36b476c57e3b03050f538c6.

DOD-02 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	projects/DODO%20LimitOrder/contracts/DODOLimitOrder.sol (57f4cad): 56~60, 168~170	Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected errors as below:

```
56  function init(address owner, address dodoApproveProxy, address feeReciver) external
{
57      require(address(0) != dodoApproveProxy, "set dodo approve proxy to the zero
address");
58      require(address(0) != feeReciver, "set fee receiver to the zero address");
59
60      initOwner(owner);
61      _DODO_APPROVE_PROXY_ = dodoApproveProxy;
62      _FEE_RECEIVER_ = feeReciver;
63  }
```

```
168 function changeFeeReceiver (address newFeeReceiver) public onlyOwner {
169     require(address(0) != newFeeReceiver, "set fee receiver to the zero address");
170     _FEE_RECEIVER_ = newFeeReceiver;
171 }
```

Alleviation

No alleviation.

DOO-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	Informational	projects/DODO%20LimitOrder/contracts/DODOLimitOrderBot.sol (57f4cad): 105~108, 41~44	Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected errors as below:

```
41  function init(
42      address owner,
43      address dodoLimitOrder,
44      address tokenReceiver,
45      address dodoApprove
46  ) external {
47      require(address(0) != owner, "set owner to the zero address");
48      require(address(0) != dodoLimitOrder, "set dodo limit order to the zero
address");
49      require(address(0) != tokenReceiver, "set token receiver to the zero address");
50      require(address(0) != dodoApprove, "set dodo approve to the zero address");
51
52      initOwner(owner);
53      _DODO_LIMIT_ORDER_ = dodoLimitOrder;
54      _TOKEN_RECEIVER_ = tokenReceiver;
55      _DODO_APPROVE_ = dodoApprove;
56  }
```

```
105  function changeTokenReceiver(address newTokenReceiver) external onlyOwner {
106      require(address(0) != newTokenReceiver, "DODOLimitOrderBot: set token receiver
to the zero address");
107      _TOKEN_RECEIVER_ = newTokenReceiver;
108      emit changeReceiver(newTokenReceiver);
109  }
```

Alleviation

No alleviation.

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.

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.