Audit of smart contracts WaterDeFi

Revision 1 dated 03.25.2021

Telescr.in

Contents

Brief information

Project: waterdefi.com
Network: Binance
Compiler version: ^0.6.12
Optimization: enabled
The audit date: 03.25.2021

Information

The contract code was reviewed and analyzed for vulnerabilities, logical errors and developer exit scams possibilities. This work was carried out concerning the project source code and documentation provided by the customer.

During the audit neither errors that can affect the security of funds nor exit scam possibility was detected.

General conclusion

As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam were found. No backdoors found either.

Telescr.in guarantees the safety and performance of the WaterDeFi contract.

Liability disclaimer

The telescr.in team within this audit framework is not responsible for the developers or third parties' actions on the platforms associated with this project (websites, mobile applications, and so on). The audit confirms and guarantees only the smart contract correct functioning in the revision provided by the project developers.

Confirmed by digital signature

Aggregated data

The Contract analysis was performed using the following methods:

- Static analysis
  - Checking the code for common errors leading to the most common vulnerabilities
- Dynamic analysis
  - The Contract launching and carrying out the attacks various kinds to identify vulnerabilities
- Code Review

Received data

Recommendation	**Type**	**Priority**	**Probability of occurrence**	**Line**
Incorrect minimal compiler version in pragma	Error	Low	N/a	7
Minor issue in transfer to excluded account	Remark	Low	Medium	638-647
Functions returning constant may be marked as pure	Remark	Low	N/a	487, 491, 495, 718, 722
Transferring amount includes commissions	Remark	Low	N/a	607-626
Unreachable branch in condition statement	Remark	Low	N/a	624
Unused method from Address library	Remark	Low	N/a	282-386
No check for amount in transfer method	Remark	Low	N/a	607-626

A. Errors

1. Incorrect minimal compiler version in pragma

Line 7
Contract is uncompilable with solc 0.6.0 because of Address library. We used 0.6.12 to compile and verify it.

B. Warnings

Not found.

C. Notice

Not found.

D. Remarks

1. Minor issue in transfer to excluded account

Line 638-647
Transfer to excluded accounts is performed in both tokens and reflections. This may lead to minor issue when token amount changes depending on inclusion/exclusion account.
According to common sense exclusion operation is exceptional action so it seems that this issue doesn’t affect system too much.

2. Functions returning constant may be marked as pure

Lines 487, 491, 495, 718, 722

```
Functions
        function name()
        function symbol()
        function decimals()
        function _getTaxFee()
        function _getMaxTxAmount()
```

do not read contract state returning constant values and can be marked as pure.

3. Transferring amount includes commissions

Line 607-626
Amount that user is trying to transfer includes 10% commissions, so if one wants to transfer 100 Water, one need to remember that recipient receives only 90.

4. Unreachable branch in condition statement

Line 624
In transfer function `else` condition is excess and unreachable

5. Unused methods from Address library

Line 282-386
Address library is used only for checking if address is a contract. Other methods are not used and can be cleared from code.

6. No check for amount in transfer method

Line 607-626
Although checks are performed later in SafeMath methods and doesn't cause error, it would be nice to check it in the beginning of transfer method and display more informative error message.

Application. Error classification

Priority	
informational	This question is not directly related to functionality but may be important to understand.
low	This question has nothing to do with security, but it can affect some behavior in unexpected ways.
Average	The problem affects some functionality but does not result in an economically significant user funds loss.
high	This issue can result in the user funds loss.
Probability	
low	It is unlikely that the system is in a state in which an error could occur or could be caused by any party.
Average	This problem may likely arise or be caused by some party.
high	It is highly likely that this problem could arise or could be exploited by some parties.

Application. Digital bytecode print

The audit was carried out for the code certain version on the compiler version ^0.6.12 with the optimization enabled.

To check the contract bytecode for identity to the one that was analyzed during the audit, you must:

1. Get contract bytecode (in any block explorer)
2. Get SHA1 from bytecode string
3. Compare with reference in this report

Sha1 from bytecode:
6c3a7d39eb30d94e05fc1a9740bc7d742d98f876
Sha1 from bytecode (non-metadata):
84ddd30c6ceb9212759e1d4481ae153c7aae3886
Contract address:
0x57f81252d1187754048f5af1938226b9034b599f

Check the digital print

Application. Signature of the audit report

```
{
  "address": "0x505ade8cea4db608250e503a5e8d4cb436044d2e",
  "msg": "As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam were found. No backdoors found either. Telescr.in guarantees the safety and performance of the WaterDeFi contract. Sha1 from bytecode: 6c3a7d39eb30d94e05fc1a9740bc7d742d98f876 Sha1 from bytecode (non-metadata): 84ddd30c6ceb9212759e1d4481ae153c7aae3886 Contract address: 0x57f81252d1187754048f5af1938226b9034b599f",
  "sig": "0x9d0f0448763ba2dae5c5786b6527abd64bff74b06b6ad3ff7b34544e68fce8f62ce7e3f6120df8f73e5a8d07550d5e97fae2ad4d5d9a6731d9dd3cd95196d1fe1b",
  "version": "3"
}
```

Check the signature