CYPHEX

Smart Contract Security Audit

Audit Details:

Audited project:	Inflow Token
Deployer address:	0xbB8147C106Fb64E93f74Ec38ecC3e09aDf4F5A25
Blockchain:	Polygon Mumbai Testnet
Project website:	Not provided

DATE

28.06.2022

CYPHEX

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Cyphex and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Cyphex) owe no duty of care towards you or any other person, nor does Cyphex make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Cyphex hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Cyphex hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Cyphex, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

Cyphex was commissioned by Inflow Project Developer to perform an audit of smart contracts:

- https://mumbai.polygonscan.com/address/0x5CBC4F6e4428b2e4814f3cf3999635eBd589302F
- https://mumbai.polygonscan.com/address/0xFAaA764Bd35e61b5dA76A0d6AF5334E42F50a28e
- https://mumbai.polygonscan.com/address/0x125b3132b3c30912f15f5f1e380e936342c08654

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract. The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.
- Ensure that the smart contract logic and functionalities fit.

CYPHEX

Inflow Token Contract details

Contract details at 28.06.2022.

Contract name:	**InflowToken**
Compiler version:	**v0.8.9+commit.e5eed63a**
Contract address:	0x5CBC4F6e4428b2e4814f3cf3999635eBd589302F
Total supply:	Unlimited
Token ticker:	FLW
Decimals:	18
Token holders:	5
Transactions count:	12
Top 100 holders dominance:	100%
Contract deployer address:	0xbb8147c106fb64e93f74ec38ecc3e09adf4f5a25
Contract's current owner address:	0xbb8147c106fb64e93f74ec38ecc3e09adf4f5a25

Top 5 token holders

Rank	Address	Quantity	Percentage	Analytics
1	0xfaaa764bd35e61b5da76a0d6af5334e42f50a28e	7,499,996.839689748230819841	60.0000%	
2	0xcc8772b133452d265d4a2152453a61ccc66e2a75	5,000,000	40.0000%	
3	0x094b24ad3c432bd3b766297831c5f9fb66c7903a	1.166666666666666666	0.0000%	
4	0xa81512d54d4691be99c008edc9a19d89073eaf54	1.033333333333333333	0.0000%	
5	0x69a9fbb18c7af8dd56b0f7c49c45aba2083302ea	0.96031025176918016	0.0000%	

Price Consumer Contract details

Contract details at 28.06.2022.

Contract name:	**PriceConsumerV3**
Compiler version:	**v0.8.9+commit.e5eed63a**
Contract address:	0x125b3132b3c30912F15F5F1e380e936342C08654
Deployer address:	0xbb8147c106fb64e93f74ec38ecc3e09adf4f5a25
Contract owner address:	-

Private Sale Contract details

Contract details at 28.06.2022.

Contract name:	**InflowPrivateSale**
Compiler version:	**v0.8.9+commit.e5eed63a**
Contract address:	0xFAaA764Bd35e61b5dA76A0d6AF5334E42F50a28e
Deployer address:	0xbb8147c106fb64e93f74ec38ecc3e09adf4f5a25
Contract owner address:	0xbb8147c106fb64e93f74ec38ecc3e09adf4f5a25

Issues Checking Status

No	Issue description.	Checking status
1	Compiler errors.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions	Passed
3	Possible delays in data delivery.	Low Severity
4	Oracle calls.	Medium Issues
5	Front running.	Passed
6	Timestamp dependence.	Medium Issues
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	High Severity
11	Economy model of the contract.	Low Severity
12	The impact of the exchange rate on the logic.	High Severity
13	Private user data leaks.	Passed
14	Malicious Event log.	Medium Issues
15	Scoping and Declarations.	Low Severity
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Medium Issues
19	Cross-function race conditions.	Medium Issues
20	Safe Open Zeppelin contracts implementation and usage.	Passed
21	Fallback function security.	Passed

Security Issues

High Severity Issues

InflowPrivateSale.sol

- Line 138: We assumed that '_maticUsd' parameter is used to get latest price of matic. It can be entered by user who calls 'buyTokens' function. If parameter set higher than MATIC price. User will be able to buy tokens with close to zero price.

InflowToken.sol

- Owner can mint new tokens with calling mintToken function anytime he/she wants
- If owner private key stolen, the exploiter can mint tokens. Recommended to transfer ownership to the multi-sig wallet.

Medium Severity Issues

InflowPrivateSale.sol

- Line 345 - Line 391: Since release method called by revoke method. There is double substitution on 'vestingSchedulesTotalAmount' variable.
- If owner of InflowToken.sol and InflowPrivateSale.sol contracts will be same. The Private sale contract should sign newly added users messages off-chain to let them be able to buy the token, and that means owner key can be accesible from off-chain oracle or a person. Vulnerability on off-chain oracle can also cause minting infinite tokens. Recommendation is use seperate wallets for signing ECDSA and minting Inflow Tokens

Low Severity Issues

InflowPrivateSale.sol

- Line 46: Variable named 'usedWhitelistSigs' has no visibility argument.
- Line 513: Block.timestamp can be abused by block miners. Using block.number is much secure.
- Line 112: Instead of checking 'token' parameters address. You can change parameter type of token from 'address' to 'IERC20'.

Owner Privileges for whole Project

- **Owner can mint unlimited number of tokens.**
- **Owner can change state of private sale.**
- **Owner of PrivateSale contract can sign messages for buying new tokens through private sale contract.**

Conclusion

Smart contracts contain **High**, medium and low severity issues.

Recommendations

- Do not set same address as owner for both Private sale contract and token contract. Because private sale contract owner key is probably will be used on off-chain system that signs new users to let them be able to join private sale. If this key stolen, new tokens can be minted.
- Change owner to operator for message signing for private sale contract.
- Private sale contract owner should only withdraw tokens instead off-chain message signing.
- Token minting should be limited or the owner of token should be transferred to multi-sig wallet.
- Instead of defining variable as private and create a getter view function, variable can just be defined as public.
- Returning value from non callable from contract function is unnecessary.
- Instead of setting inflowtoken price in function. Set it as global constant variable.
- Setting admin wallet variable as private will not protect 3rd party from reading that value. It can be reachable with contract bytecode at fourth storage slot.
- Instead of using off-chain oracle fed smart contract, get direct token price from highest liquidity locked DeFi will be much more accurate on-chain.
- Adding require statement that checks the caller is owner and also the caller is contract.
- Since 'withdraw' method is onlyOwner, you can send tokens directly to the msg.sender instead of owner().
- The initial value of boolean variable is already false, so redefining as false is non sense here.

Cyphex note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.