GORDIAN AGENCY

SMART CONTRACT SECURITY AUDIT

August 7th, 2021

Professional Auditing Agency

Website https://gordian.agency/

AUDIT DETAILS

Gordian Agency

Project

Iagon

Deployer Address

0xfdd8a2ffb5e3676eac72fa83a2de032cfb372747

Client Contacts

@NavjitDhaLiwaL

Blockchain

Ethereum

Project Website

https://www.iagon.com/

BACKGROUND

Gordian Agency

Gordian Agency was commissioned by Iagon to perform an updated audit of smart contracts:

1. OLD_IAGON Token: [IagonToken.sol]
2. NEW_IAGON Token [IAGONNEW.sol]
3. Burner: [Burner.sol]
4. Swapper: [Swapper.sol]

The purpose of this audit was to achieve the following:

- Ensure that the token contract functions as intended
- Identify potential security issues with the token contracts

The information in this report should be used to understand the risk exposure of the smart contracts, and as a guide to improve the security posture of the smart contracts by remediating the issues that were identified.

GORDIAN CERTIFIED AGENCY

IAGON Token Smart Contract Details

Smart Contract Testing and Audit

Swapper contract is deployed on the Ethereum Mainnet to swap the OLD_IAGON TOKEN to NEW_IAGON TOKEN. Four Smart contracts listed below are deployed and Burner and Swapper Contract were tested on the Rinkeby testnet to check its usage and security vulnerabilities with manual checks.

1. OLD_IAGON Token: IagonToken.sol
2. NEW_IAGON Token IAGONNEW.sol
3. Burner: Burner.sol
4. Swapper: Swapper.sol

Methodology

1. **White Box Testing / Source Code Review**

   Understanding programming language used and coding practices followed is a key in
   performing an efficient source code review solution. It is performed by Understand Program Specification, Obtain and Review Source Code and identifying flaws

2. **Black Box Testing**

   Understanding how your system works is a key in providing you the right security
   Solution. Multiple Exploitations and garbage collection data is introduced to the contract to check its response.

Contract Summary

1. **Contract Context (abstract)**
   - From Context
     - _msgData() (internal)
     - _msgSender() (internal)
2. **Contract IAGON (Inherits ERC20, ECR20Burnable, ERC20Snapshot, AccessControl, ERC20Permit)**
   - From IAGON
     - constructor() (internal)
     - snapshot() (public)
     - _beforeTokenTransfer() (internal)
3. **Contract ERC20 (Inherits Context, IERC20, IERC20Metadata)**
   - From ERC20
     - constructor(string memory name_, string memory symbol_) (internal)
     - name() (public)
     - symbol() (public)
     - decimals() (public)
     - balanceOf(address account) (public)
     - name() (external)
     - totalSupply() (public)
     - transfer(address recipient, uint256 amount) (public)
     - transferFrom(address sender, address recipient, uint256 amount) (public)

Contract Summary

3. **Contract ERC20 (Inherits Context, IERC20, IERC20Metadata)**
    - From ERC20
        - allowance(address owner, address spender) (public)
        - approve(address spender, uint256 amount) (public)
        - increaseAllowance(address spender, uint256 addedValue) (public)
        - decreaseAllowance(address spender, uint256 subtractedValue) (public)
        - _transfer(address sender, address recipient, uint256 amount) (internal)
        - _mint(address account, uint256 amount) (internal)
        - _burn(address account, uint256 amount) (internal)
        - _approve(address owner, address spender, uint256 amount) (internal)
        - _beforeTokenTransfer(address from, address to, uint256 amount) (internal)

4. **Abstract Contract ERC20Burnable (Inherits Context, IERC20)**
    - From ERC20Burnable
        - burn(uint256 amount) (public)
        - burnFrom(address account, uint256 amount) (public)

Contract Summary

5. **Abstract Contract ERC20Snapshot (Inherits IERC20)**
   - From ERC20Snapshot
     - _snapshot() (internal)
     - balanceOfAt(address account, uint256 snapshotId) (public)
     - totalSupplyAt(uint256 snapshotId) (public)
     - _beforeTokenTransfer(address from, address to, uint256 amount) (internal)
     - _valueAt(uint256 snapshotId, Snapshots storage snapshots) (private)
     - _updateAccountSnapshot(address account) (private)
     - _updateTotalSupplySnapshot() (private)
     - _updateSnapshot(Snapshots storage snapshots, uint256 currentValue) (private)
     - _lastSnapshotId(uint256[] storage ids) (private)

6. **Abstract Contract Multicall**
   - From Multicall
     - multicall(bytes[] calldata data) (external)
7. **Abstract Contract ERC165 (Inherits IERC165)**
   - From ERC165
     - supportsInterface(bytes4 interfaceId) (public)

Findings

Manual Checks

Swapper Contract

1. Old Tokens can only be swapped after approving swapper contract as its spender with spending amount
2. Token cannot be swapped if approvingDeadline is crossed
3. Only Owner of Contract can change the ApprovalDeadLine Of Swap
4. Burner contract should be provided the Ownership of OldToken in order to swap
5. Swapper Contract needs to have enough NewTokens in order to swap.
6. OldToken will be transferred to the Burner contract on each swap execution
7. Burner Contract will burn the received contract only if burner has the ownership of OldToken contract
8. On every Swap all the OldToken are burned and exact amounts of NewTokens are allocated for the address who executed the swap
9. 10% of the NewTokens will be transferred to the address who executed the swap instantly after the swap
10. Remaining 90% of the NewToken can only be extracted from the swapper contract after 3 months with increments of 7.5% of the total amount of tokens allocated, over a 12 month period.
11. Vested amount of each user can be checked anytime publicly.
12. Unlocked Amount can be checked anytime publicly.
13. Once the NewTokens is unlocked it can be claimed anytime within a 12 months + 2 weeks time frame ,starting from the time swap was executed.
14. Tokens can only be swapped within a 12 months + 2 weeks time frame, starting from the time swap was executed.
15. Owner of Swapper Contract can emergency withdraw any amount of New IAGON Token from Swapper Contract

`

ISSUES DETAILS

Burner Contract

1. Burner Contract can burn Old IAGON Token once the ownership of Old IAGON Token is provided.
2. The Ownership Old IAGON Token can be transferred from Burner to any Address by Burner Contract Owner

Vulnerability Analysis

Swapper Contract

1. All the External Contract address are made constants and cannot be changed preventing Reentrancy Attacks
2. Guard Check on SwapNow() method to verify the input parameters
3. Guard Check on emergencyWithdraw() and updateApprovalDeadline() method to verify the input parameters
4. Emergency Stop is implemented as emergencyWithdraw() method to disable critical contract functionality in case of an emergency.

Burner Contract

1. Guard Check on returnTokenOwnership() method to verify the contract caller

CONCLUSION

The audited contract contains no issues and is safe to deploy.

NOTES:

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

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 Gordian and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Gordian) owe no duty of care towards you or any other person, nor does Gordian 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 Gordian 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, Gordian hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Gordian, 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.