Guard

Smart contracts security assessment

Final report

Tariff: Standard

HPO Token

March 2024

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for **HPO Token**.

Hippo Token is an ERC-20 standard token with ERC20Burnable and ERC20Pausable extensions from the OpenZeppelin repository. The token has the minting functionality and no taxes. Mint is managed by accounts with the `MINTER_ROLE`, governed by the project owner. The total mintable amount at any moment can't exceed the `totalRelease` value, set by the owner. The total minted amount can't exceed the `maxTotalSupply` value. Initial mint is about 1/3 of the `maxTotalSupply`.

The code is available at the @hippowallet/hippotoken GitHub repository and was audited after the commit 649f50c8601f56750efdd38f677619c732502b9d.

Name	HPO Token
Audit date	2024-02-27 - 2024-03-01
Language	Solidity
Platform	Binance Smart Chain

Contracts checked

Name	Address
ERC20PresetMinterPauser	
HPOToken	

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	not passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed

Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	not passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Privileged functions (ERC20PresetMinterPauser)

Status: Open

The token can be minted by the project owner or with his permission.

All transfer related token operations (`transfer`, `transferFrom`, `mint`, `burn`, `burnFrom`) can be paused by the project owner or with his permission.

Recommendation: Secure the owner account and all other accounts with the `DEFAULT_ADMIN_ROLE`.

Team response: The inclusion of a token pausing functionality within our contract is not intended for the selective suspension of individual wallet activities. Rather, it is designed with a holistic approach that allows for the temporary cessation of all token transactions, inclusive of those pertaining to our organization's own tokens. This ensures an equitable impact across all users, emphasizing a shared commitment to safeguarding interests without introducing specific risks to any party.

This feature's integration is a direct response to the stringent regulatory requirements imposed by our jurisdiction of operation. Such measures are vital for providing a framework that enables us to effectively manage and mitigate potential risks associated with financial crimes, including but not limited to money laundering. It is imperative to understand that this mechanism is employed as a protective measure, designed to ensure compliance and safeguard the integrity of all transactions within the ecosystem.

Consequently, the implementation of this feature does not entail any inherent risk to users. It is established as a precautionary measure, aligned with regulatory obligations and committed to the collective security and trust of our platform's participants. Furthermore, it is important to note that the nature of this feature's implementation is such that it underscores an intention for it to remain inactive under normal operating conditions. This deliberate design choice clearly indicates that the feature was written with the foresight of not being implemented, serving only as a regulatory compliance measure to be activated under strictly defined circumstances. This approach underscores our

dedication to maintaining a transparent, secure, and compliant operational environment.

Medium severity issues

No issues were found

Low severity issues

1. Wrong AccessControl modifier (HPOToken)

Status: Open

The function release is restricted to the accounts assigned an admin role of the DEFAULT_ADMIN_ROLE role, which is by coincidence is the DEFAULT_ADMIN_ROLE itself.

```
    function release(uint256 amount) public onlyRole(getRoleAdmin(DEFAULT_ADMIN_ROLE))
returns (bool) {
        require((_maxTotalSupply - _totalRelease) >= amount, "Release amount out of max
total supply");
        _totalRelease += amount;
        return true;
    }
```

Recommendation: Correct usage is `onlyRole(DEFAULT_ADMIN_ROLE)`.

Conclusion

HPO Token ERC20PresetMinterPauser, HPOToken contracts were audited. 1 high, 1 low severity issues were found.

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 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 without 0xGuard prior written consent.

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 0xGuard 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.

Slither output

```
INFO:Detectors:
HPOToken._maxTotalSupply (contracts/HPOToken.sol#10) is set pre-construction with a non-constant function or state variable:
        - 30_000_000_000 * (10 ** _decimals)
HPOToken._initialRelease (contracts/HPOToken.sol#11) is set pre-construction with a non-constant function or state variable:
        - 10_000_000_000 * (10 ** _decimals)
HPOToken._initial_bridge_Release (contracts/HPOToken.sol#12) is set pre-construction with a non-constant function or state variable:
        - 500_000 * (10 ** _decimals)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state
INFO:Detectors:
Pragma version0.8.11 (contracts/ERC20PresetMinterPauser.sol#4) allows old versions
Pragma version0.8.11 (contracts/HPOToken.sol#3) allows old versions
solc-0.8.11 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Variable HPOToken._initialRelease (contracts/HPOToken.sol#11) is not in mixedCase
Variable HPOToken._initial_bridge_Release (contracts/HPOToken.sol#12) is not in mixedCase
Modifier HPOToken._isReleased(uint256) (contracts/HPOToken.sol#15-19) is not in mixedCase
Modifier HPOToken._maxTotalSupplyNotFull(uint256) (contracts/HPOToken.sol#21-25) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
HPOToken._decimals (contracts/HPOToken.sol#9) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
HPOToken._initialRelease (contracts/HPOToken.sol#11) should be immutable
HPOToken._initial_bridge_Release (contracts/HPOToken.sol#12) should be immutable
HPOToken._maxTotalSupply (contracts/HPOToken.sol#10) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-
```

```
variables-that-could-be-declared-immutable
INFO:Slither:. analyzed (36 contracts with 88 detectors), 15 result(s) found
```

0x Guard