0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

PulseLatina

June 2024

0xguard.com

hello@0xguard.com

Contents

1. Introduction 3
2. Contracts checked 3
3. Procedure 3
4. Known vulnerabilities checked 4
5. Classification of issue severity 5
6. Issues 5
7. Conclusion 8
8. Disclaimer 9
9. Slither output 10
10. Slither ERC conformance 22

Introduction

The report has been prepared for **PulseLatina**.

The audited project is MasterChef farm minting ERC20 token as a reward. Reward token is not capped, and the emission rate is limited to 50e18 per block. Pools of the MasterChef contract have both deposit and withdraw fee. Deposit fee can be set up to 100%, withdraw fee can be set to an arbitrary value.

The code is available at the GitHub repository and was audited after the commit 06bb3c248c9d4ebdd2948c2ee9c70a77a86d0c32.

Report Update.

The contract's code was updated according to this report and rechecked after the commit 326fd8de046bdb931a9897f6b78ee9f194a037ce.

Name	PulseLatina
Audit date	2024-06-23 - 2024-06-25
Language	Solidity
Platform	Pulse Chain

Contracts checked

Name	Address
MasterChef	

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	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	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Outdated Compiler Version	passed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. Unlimited emission rate (MasterChef)

Status: Fixed

The emission rate of the reward token is not capped. The project owner can mint any amount of reward tokens in a single block.

Recommendation: Secure ownership by using Timelock and Multisig. Limit emission rate in the `updateEmissionRate` function.

2. Withdraw fee can be set beyond 100% (MasterChef)

Status: Fixed

The project owner can set withdraw fee for certain pools beyond 100% resulting in reverting all withdrawal attempts.

Recommendation: Add withdraw fee limit or add emergency withdraw function.

Medium severity issues

1. Withdraw fee can be avoided by users (MasterChef)

Status: Fixed

Withdraw fee is charged within the `withdraw` function. Users of the project can claim their rewards by calling `deposit(pid,0)` or `withdraw(pid,0,0)`, then call `emergencyWithdraw(pid)`, and finally deposit back any amount in case of partial withdrawal.

Recommendation: Remove withdraw fee.

Low severity issues

1. Unused code (MasterChef)

Status: Partially fixed

The `devaddr` variable and `AddNAddressesForTestingStability`, `AddressExistsInDepositAddressesMapping` functions are not used.

Recommendation: Remove unused code.

Conclusion

PulseLatina MasterChef contract was audited. 2 high, 1 medium, 1 low severity issues were found.

2 high, 1 medium severity issues have been fixed in the update.

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:
MasterChef.safeLatinaTransfer(address,uint256) (contracts/
PulseLatinaFarm.sol#1805-1812) ignores return value by latina.transfer(_to,latinaBal)
(contracts/PulseLatinaFarm.sol#1808)
MasterChef.safeLatinaTransfer(address,uint256) (contracts/
PulseLatinaFarm.sol#1805-1812) ignores return value by latina.transfer(_to,_amount)
(contracts/PulseLatinaFarm.sol#1810)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-
transfer
INFO:Detectors:
MasterChef.pendingLatina(uint256,address) (contracts/PulseLatinaFarm.sol#1553-1575)
performs a multiplication on the result of a division:
        - latinaReward =
multiplier.mul(latinaPerBlock).mul(pool.allocPoint).div(totalAllocPoint) (contracts/
PulseLatinaFarm.sol#1566-1569)
        - accLatinaPerShare =
accLatinaPerShare.add(latinaReward.mul(1e12).div(lpSupply)) (contracts/
PulseLatinaFarm.sol#1570-1572)
MasterChef.updatePool(uint256) (contracts/PulseLatinaFarm.sol#1586-1607) performs a
multiplication on the result of a division:
        - latinaReward =
multiplier.mul(latinaPerBlock).mul(pool.allocPoint).div(totalAllocPoint) (contracts/
PulseLatinaFarm.sol#1597-1600)
        - pool.accLatinaPerShare =
pool.accLatinaPerShare.add(latinaReward.mul(1e12).div(lpSupply)) (contracts/
PulseLatinaFarm.sol#1603-1605)
MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735) performs a multiplication on the result of a division:
        - depositFee = _amount.mul(pool.depositFeeBP).div(10000) (contracts/
PulseLatinaFarm.sol#1716)
        - feeAddress1Share = depositFee.mul(50).div(100) (contracts/
PulseLatinaFarm.sol#1717)
MasterChef.withdraw(uint256,uint256,address) (contracts/PulseLatinaFarm.sol#1738-1791)
performs a multiplication on the result of a division:
        - withdrawFee = _amount.mul(pool.withdrawFeeBP).div(10000) (contracts/
PulseLatinaFarm.sol#1761)
        - feeAddressShare = withdrawFee.mul(50).div(100) (contracts/
PulseLatinaFarm.sol#1762)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
LatinaToken._writeCheckpoint(address,uint32,uint256,uint256) (contracts/PulseLatinaFarm.sol#1344-1369) uses a dangerous strict equality:
        - nCheckpoints > 0 && checkpoints[delegatee][nCheckpoints - 1].fromBlock == blockNumber (contracts/PulseLatinaFarm.sol#1356-1357)
MasterChef.updatePool(uint256) (contracts/PulseLatinaFarm.sol#1586-1607) uses a dangerous strict equality:
        - lpSupply == 0 || pool.allocPoint == 0 (contracts/PulseLatinaFarm.sol#1592)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
Reentrancy in MasterChef.deposit(uint256,uint256) (contracts/PulseLatinaFarm.sol#1629-1673):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1640)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1643)
                - latinaPerBlock = latinaPerBlock / 2 (contracts/PulseLatinaFarm.sol#1617)
        MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431) can be used in cross function reentrancies:
        - MasterChef.constructor(LatinaToken,address,address,uint256,uint256,uint256) (contracts/PulseLatinaFarm.sol#1469-1485)
        - MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431)
        - MasterChef.pendingLatina(uint256,address) (contracts/PulseLatinaFarm.sol#1553-1575)
        - MasterChef.updateEmissionRate(uint256) (contracts/PulseLatinaFarm.sol#1825-1828)
        - MasterChef.updatePool(uint256) (contracts/PulseLatinaFarm.sol#1586-1607)
Reentrancy in MasterChef.deposit(uint256,uint256) (contracts/PulseLatinaFarm.sol#1629-1673):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1640)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1646)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount) (contracts/PulseLatinaFarm.sol#1650-1654)
        - pool.lpToken.safeTransfer(feeAddress,depositFee) (contracts/PulseLatinaFarm.sol#1664)
```

```
        State variables written after the call(s):
        - user.amount = user.amount.add(_amount).sub(depositFee) (contracts/
PulseLatinaFarm.sol#1666)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reentrancy in MasterChef.deposit(uint256,uint256) (contracts/
PulseLatinaFarm.sol#1629-1673):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1640)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1646)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount)
(contracts/PulseLatinaFarm.sol#1650-1654)
        State variables written after the call(s):
        - user.amount = user.amount.add(_amount) (contracts/PulseLatinaFarm.sol#1668)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reentrancy in MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1694)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1697)
                - latinaPerBlock = latinaPerBlock / 2 (contracts/
PulseLatinaFarm.sol#1617)
        MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431) can be used in
cross function reentrancies:
        - MasterChef.constructor(LatinaToken,address,address,uint256,uint256,uint256)
(contracts/PulseLatinaFarm.sol#1469-1485)
        - MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431)
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.updateEmissionRate(uint256) (contracts/
PulseLatinaFarm.sol#1825-1828)
```

```
        - MasterChef.updatePool(uint256) (contracts/PulseLatinaFarm.sol#1586-1607)
Reentrancy in MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1694)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1700)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount)
(contracts/PulseLatinaFarm.sol#1704-1708)
        - pool.lpToken.safeTransfer(referral,depositFee) (contracts/
PulseLatinaFarm.sol#1722)
        - pool.lpToken.safeTransfer(feeAddress,feeAddress1Share) (contracts/
PulseLatinaFarm.sol#1724)
        - pool.lpToken.safeTransfer(referral,feeAddress2Share) (contracts/
PulseLatinaFarm.sol#1725)
        State variables written after the call(s):
        - user.amount = user.amount.add(_amount).sub(depositFee) (contracts/
PulseLatinaFarm.sol#1728)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
        - user.rewardDebt = user.amount.mul(pool.accLatinaPerShare).div(1e12)
(contracts/PulseLatinaFarm.sol#1733)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reentrancy in MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1694)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1700)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount)
(contracts/PulseLatinaFarm.sol#1704-1708)
        State variables written after the call(s):
```

```
        - user.amount = user.amount.add(_amount) (contracts/PulseLatinaFarm.sol#1730)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reentrancy in MasterChef.withdraw(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1738-1791):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1747)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1750)
                - latinaPerBlock = latinaPerBlock / 2 (contracts/
PulseLatinaFarm.sol#1617)
        MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431) can be used in
cross function reentrancies:
        - MasterChef.constructor(LatinaToken,address,address,uint256,uint256,uint256)
(contracts/PulseLatinaFarm.sol#1469-1485)
        - MasterChef.latinaPerBlock (contracts/PulseLatinaFarm.sol#1431)
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.updateEmissionRate(uint256) (contracts/
PulseLatinaFarm.sol#1825-1828)
        - MasterChef.updatePool(uint256) (contracts/PulseLatinaFarm.sol#1586-1607)
Reentrancy in MasterChef.withdraw(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1738-1791):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1747)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1753)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        State variables written after the call(s):
        - user.amount = user.amount.sub(_amount) (contracts/PulseLatinaFarm.sol#1756)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reentrancy in MasterChef.withdraw(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1738-1791):
        External calls:
```

```
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1747)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1753)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransfer(address(msg.sender),_amount) (contracts/
PulseLatinaFarm.sol#1759)
        - pool.lpToken.safeTransfer(feeAddress,withdrawFee) (contracts/
PulseLatinaFarm.sol#1769)
        - pool.lpToken.safeTransfer(referralAddress,withdrawFee) (contracts/
PulseLatinaFarm.sol#1772)
        - pool.lpToken.safeTransfer(feeAddress,withdrawFee) (contracts/
PulseLatinaFarm.sol#1778)
        - pool.lpToken.safeTransfer(feeAddress,feeAddressShare) (contracts/
PulseLatinaFarm.sol#1781)
        - pool.lpToken.safeTransfer(referralAddress,referralAddressShare) (contracts/
PulseLatinaFarm.sol#1782)
        - pool.lpToken.safeTransfer(address(msg.sender),_amount - withdrawFee)
(contracts/PulseLatinaFarm.sol#1786)
        State variables written after the call(s):
        - user.rewardDebt = user.amount.mul(pool.accLatinaPerShare).div(1e12)
(contracts/PulseLatinaFarm.sol#1789)
        MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440) can be used in cross
function reentrancies:
        - MasterChef.pendingLatina(uint256,address) (contracts/
PulseLatinaFarm.sol#1553-1575)
        - MasterChef.userInfo (contracts/PulseLatinaFarm.sol#1440)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1
INFO:Detectors:
MasterChef.add(uint256,IERC20,uint16,uint16,bool) (contracts/
PulseLatinaFarm.sol#1493-1521) should emit an event for:
        - totalAllocPoint = totalAllocPoint.add(_allocPoint) (contracts/
PulseLatinaFarm.sol#1510)
MasterChef.set(uint256,uint256,uint16,bool) (contracts/PulseLatinaFarm.sol#1524-1542)
should emit an event for:
        - totalAllocPoint =
totalAllocPoint.sub(poolInfo[_pid].allocPoint).add(_allocPoint) (contracts/
PulseLatinaFarm.sol#1537-1539)
MasterChef.updateEmissionRate(uint256) (contracts/PulseLatinaFarm.sol#1825-1828) should
emit an event for:
        - latinaPerBlock = _latinaPerBlock (contracts/PulseLatinaFarm.sol#1827)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-
arithmetic
INFO:Detectors:
MasterChef.constructor(LatinaToken,address,address,uint256,uint256,uint256)._devaddr
(contracts/PulseLatinaFarm.sol#1471) lacks a zero-check on :
                - devaddr = _devaddr (contracts/PulseLatinaFarm.sol#1478)
MasterChef.constructor(LatinaToken,address,address,uint256,uint256,uint256)._feeAddress1
 (contracts/PulseLatinaFarm.sol#1472) lacks a zero-check on :
                - feeAddress = _feeAddress1 (contracts/PulseLatinaFarm.sol#1479)
MasterChef.dev(address)._devaddr (contracts/PulseLatinaFarm.sol#1815) lacks a zero-
check on :
                - devaddr = _devaddr (contracts/PulseLatinaFarm.sol#1817)
MasterChef.setFeeAddress1(address)._feeAddress1 (contracts/PulseLatinaFarm.sol#1820)
lacks a zero-check on :
                - feeAddress = _feeAddress1 (contracts/PulseLatinaFarm.sol#1822)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-
address-validation
INFO:Detectors:
Reentrancy in MasterChef.deposit(uint256,uint256) (contracts/
PulseLatinaFarm.sol#1629-1673):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1640)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1643)
                - currentHalvingNumber ++ (contracts/PulseLatinaFarm.sol#1615)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1643)
                - firstHalvingHappened = true (contracts/PulseLatinaFarm.sol#1614)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1643)
                - latinaLeftToBeMintedThisHalving = maxLatinaMintedPerHalving -
latinaMintedThisHalving (contracts/PulseLatinaFarm.sol#1624)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1643)
                - latinaMintedThisHalving = latinaMintedThisHalving + latinaMinted
(contracts/PulseLatinaFarm.sol#1611)
                - latinaMintedThisHalving = diff (contracts/PulseLatinaFarm.sol#1621)
Reentrancy in MasterChef.deposit(uint256,uint256) (contracts/
PulseLatinaFarm.sol#1629-1673):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1640)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1646)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
```

```
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount)
(contracts/PulseLatinaFarm.sol#1650-1654)
        State variables written after the call(s):
        - AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address(msg.sender))
(contracts/PulseLatinaFarm.sol#1657)
                - depositAddresses.push(addy) (contracts/PulseLatinaFarm.sol#1854)
        - AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address(msg.sender))
(contracts/PulseLatinaFarm.sol#1657)
                - depositAddressesMapping[addy] = true (contracts/
PulseLatinaFarm.sol#1853)
        - totalDeposits ++ (contracts/PulseLatinaFarm.sol#1658)
Reentrancy in MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1694)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1697)
                - currentHalvingNumber ++ (contracts/PulseLatinaFarm.sol#1615)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1697)
                - firstHalvingHappened = true (contracts/PulseLatinaFarm.sol#1614)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1697)
                - latinaLeftToBeMintedThisHalving = maxLatinaMintedPerHalving -
latinaMintedThisHalving (contracts/PulseLatinaFarm.sol#1624)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1697)
                - latinaMintedThisHalving = latinaMintedThisHalving + latinaMinted
(contracts/PulseLatinaFarm.sol#1611)
                - latinaMintedThisHalving = diff (contracts/PulseLatinaFarm.sol#1621)
Reentrancy in MasterChef.depositReferral(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1677-1735):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1694)
        - safeLatinaTransfer(msg.sender,pending) (contracts/PulseLatinaFarm.sol#1700)
                - latina.transfer(_to,latinaBal) (contracts/PulseLatinaFarm.sol#1808)
                - latina.transfer(_to,_amount) (contracts/PulseLatinaFarm.sol#1810)
        - pool.lpToken.safeTransferFrom(address(msg.sender),address(this),_amount)
(contracts/PulseLatinaFarm.sol#1704-1708)
        State variables written after the call(s):
        - AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address(msg.sender))
(contracts/PulseLatinaFarm.sol#1711)
                - depositAddresses.push(addy) (contracts/PulseLatinaFarm.sol#1854)
        - AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address(msg.sender))
```

```
(contracts/PulseLatinaFarm.sol#1711)
                - depositAddressesMapping[addy] = true (contracts/
PulseLatinaFarm.sol#1853)
        - totalDeposits ++ (contracts/PulseLatinaFarm.sol#1712)
Reentrancy in MasterChef.withdraw(uint256,uint256,address) (contracts/
PulseLatinaFarm.sol#1738-1791):
        External calls:
        - latina.mint(address(this),pending) (contracts/PulseLatinaFarm.sol#1747)
        State variables written after the call(s):
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1750)
                - currentHalvingNumber ++ (contracts/PulseLatinaFarm.sol#1615)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1750)
                - firstHalvingHappened = true (contracts/PulseLatinaFarm.sol#1614)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1750)
                - latinaLeftToBeMintedThisHalving = maxLatinaMintedPerHalving -
latinaMintedThisHalving (contracts/PulseLatinaFarm.sol#1624)
        - HandleHalvingIfNeeded(pending) (contracts/PulseLatinaFarm.sol#1750)
                - latinaMintedThisHalving = latinaMintedThisHalving + latinaMinted
(contracts/PulseLatinaFarm.sol#1611)
                - latinaMintedThisHalving = diff (contracts/PulseLatinaFarm.sol#1621)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2
INFO:Detectors:
LatinaToken.delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32) (contracts/
PulseLatinaFarm.sol#1207-1246) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= expiry,latina::delegateBySig:
signature expired) (contracts/PulseLatinaFarm.sol#1241-1244)
MasterChef.AddressExistsInDepositAddressesMapping(address) (contracts/
PulseLatinaFarm.sol#1841-1849) uses timestamp for comparisons
        Dangerous comparisons:
        - i < depositAddresses.length (contracts/PulseLatinaFarm.sol#1842)
        - depositAddresses[i] == addy (contracts/PulseLatinaFarm.sol#1844)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-
timestamp
INFO:Detectors:
LatinaToken.getChainId() (contracts/PulseLatinaFarm.sol#1379-1385) uses assembly
        - INLINE ASM (contracts/PulseLatinaFarm.sol#1381-1383)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
MasterChef.AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address) (contracts/
```

```
PulseLatinaFarm.sol#1851-1856) compares to a boolean constant:
        -depositAddressesMapping[addy] == false (contracts/PulseLatinaFarm.sol#1852)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-
equality
INFO:Detectors:
Pragma version^0.8.0 (contracts/PulseLatinaFarm.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Parameter LatinaToken.mint(address,uint256)._to (contracts/PulseLatinaFarm.sol#1129) is
not in mixedCase
Parameter LatinaToken.mint(address,uint256)._amount (contracts/
PulseLatinaFarm.sol#1129) is not in mixedCase
Variable LatinaToken._delegates (contracts/PulseLatinaFarm.sol#1141) is not in
mixedCase
Parameter MasterChef.add(uint256,IERC20,uint16,uint16,bool)._allocPoint (contracts/
PulseLatinaFarm.sol#1494) is not in mixedCase
Parameter MasterChef.add(uint256,IERC20,uint16,uint16,bool)._lpToken (contracts/
PulseLatinaFarm.sol#1495) is not in mixedCase
Parameter MasterChef.add(uint256,IERC20,uint16,uint16,bool)._depositFeeBP (contracts/
PulseLatinaFarm.sol#1496) is not in mixedCase
Parameter MasterChef.add(uint256,IERC20,uint16,uint16,bool)._withdrawFeeBP (contracts/
PulseLatinaFarm.sol#1497) is not in mixedCase
Parameter MasterChef.add(uint256,IERC20,uint16,uint16,bool)._withUpdate (contracts/
PulseLatinaFarm.sol#1498) is not in mixedCase
Parameter MasterChef.set(uint256,uint256,uint16,bool)._pid (contracts/
PulseLatinaFarm.sol#1525) is not in mixedCase
Parameter MasterChef.set(uint256,uint256,uint16,bool)._allocPoint (contracts/
PulseLatinaFarm.sol#1526) is not in mixedCase
Parameter MasterChef.set(uint256,uint256,uint16,bool)._depositFeeBP (contracts/
PulseLatinaFarm.sol#1527) is not in mixedCase
Parameter MasterChef.set(uint256,uint256,uint16,bool)._withUpdate (contracts/
PulseLatinaFarm.sol#1528) is not in mixedCase
Parameter MasterChef.getMultiplier(uint256,uint256)._from (contracts/
PulseLatinaFarm.sol#1546) is not in mixedCase
Parameter MasterChef.getMultiplier(uint256,uint256)._to (contracts/
PulseLatinaFarm.sol#1547) is not in mixedCase
Parameter MasterChef.pendingLatina(uint256,address)._pid (contracts/
PulseLatinaFarm.sol#1554) is not in mixedCase
Parameter MasterChef.pendingLatina(uint256,address)._user (contracts/
PulseLatinaFarm.sol#1555) is not in mixedCase
```

```
Parameter MasterChef.updatePool(uint256)._pid (contracts/PulseLatinaFarm.sol#1586) is
not in mixedCase
Function MasterChef.HandleHalvingIfNeeded(uint256) (contracts/
PulseLatinaFarm.sol#1610-1625) is not in mixedCase
Parameter MasterChef.deposit(uint256,uint256)._pid (contracts/PulseLatinaFarm.sol#1629)
is not in mixedCase
Parameter MasterChef.deposit(uint256,uint256)._amount (contracts/
PulseLatinaFarm.sol#1629) is not in mixedCase
Parameter MasterChef.depositReferral(uint256,uint256,address)._pid (contracts/
PulseLatinaFarm.sol#1678) is not in mixedCase
Parameter MasterChef.depositReferral(uint256,uint256,address)._amount (contracts/
PulseLatinaFarm.sol#1679) is not in mixedCase
Parameter MasterChef.withdraw(uint256,uint256,address)._pid (contracts/
PulseLatinaFarm.sol#1738) is not in mixedCase
Parameter MasterChef.withdraw(uint256,uint256,address)._amount (contracts/
PulseLatinaFarm.sol#1738) is not in mixedCase
Parameter MasterChef.emergencyWithdraw(uint256)._pid (contracts/
PulseLatinaFarm.sol#1794) is not in mixedCase
Parameter MasterChef.safeLatinaTransfer(address,uint256)._to (contracts/
PulseLatinaFarm.sol#1805) is not in mixedCase
Parameter MasterChef.safeLatinaTransfer(address,uint256)._amount (contracts/
PulseLatinaFarm.sol#1805) is not in mixedCase
Parameter MasterChef.dev(address)._devaddr (contracts/PulseLatinaFarm.sol#1815) is not
in mixedCase
Parameter MasterChef.setFeeAddress1(address)._feeAddress1 (contracts/
PulseLatinaFarm.sol#1820) is not in mixedCase
Parameter MasterChef.updateEmissionRate(uint256)._latinaPerBlock (contracts/
PulseLatinaFarm.sol#1825) is not in mixedCase
Function MasterChef.AddNAddressesForTestingStability(int256) (contracts/
PulseLatinaFarm.sol#1833-1839) is not in mixedCase
Function MasterChef.AddressExistsInDepositAddressesMapping(address) (contracts/
PulseLatinaFarm.sol#1841-1849) is not in mixedCase
Function MasterChef.AddAddressToDepositAddressesMappingAndArrayIfDoesntExist(address)
(contracts/PulseLatinaFarm.sol#1851-1856) is not in mixedCase
Function MasterChef.GetDepositAddressesRange(uint256,uint256) (contracts/
PulseLatinaFarm.sol#1862-1874) is not in mixedCase
Function MasterChef.GetDepositAddressesCount() (contracts/
PulseLatinaFarm.sol#1876-1879) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
```

```
Variable MasterChef.depositReferral(uint256,uint256,address).feeAddress1Share
(contracts/PulseLatinaFarm.sol#1717) is too similar to
MasterChef.depositReferral(uint256,uint256,address).feeAddress2Share (contracts/
PulseLatinaFarm.sol#1718)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-
too-similar
INFO:Detectors:
Loop condition `i < depositAddresses.length` (contracts/PulseLatinaFarm.sol#1842)
should use cached array length instead of referencing `length` member of the storage
array.
 Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-
length
INFO:Detectors:
MasterChef.firstHalvingDays (contracts/PulseLatinaFarm.sol#1447) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-
variables-that-could-be-declared-constant
INFO:Detectors:
MasterChef.latina (contracts/PulseLatinaFarm.sol#1427) should be immutable
MasterChef.maxLatinaMintedPerHalving (contracts/PulseLatinaFarm.sol#1448) should be
immutable
MasterChef.startBlock (contracts/PulseLatinaFarm.sol#1444) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-
variables-that-could-be-declared-immutable
```

Slither ERC conformance

```
# Check LatinaToken

## Check functions
[✓] totalSupply() is present
        [✓] totalSupply() -> (uint256) (correct return type)
        [✓] totalSupply() is view
[✓] balanceOf(address) is present
        [✓] balanceOf(address) -> (uint256) (correct return type)
        [✓] balanceOf(address) is view
[✓] transfer(address,uint256) is present
        [✓] transfer(address,uint256) -> (bool) (correct return type)
        [✓] Transfer(address,address,uint256) is emitted
[✓] transferFrom(address,address,uint256) is present
        [✓] transferFrom(address,address,uint256) -> (bool) (correct return type)
        [✓] Transfer(address,address,uint256) is emitted
[✓] approve(address,uint256) is present
        [✓] approve(address,uint256) -> (bool) (correct return type)
        [✓] Approval(address,address,uint256) is emitted
[✓] allowance(address,address) is present
        [✓] allowance(address,address) -> (uint256) (correct return type)
        [✓] allowance(address,address) is view
[✓] name() is present
        [✓] name() -> (string) (correct return type)
        [✓] name() is view
[✓] symbol() is present
        [✓] symbol() -> (string) (correct return type)
        [✓] symbol() is view
[✓] decimals() is present
        [✓] decimals() -> (uint8) (correct return type)
        [✓] decimals() is view

## Check events
[✓] Transfer(address,address,uint256) is present
        [✓] parameter 0 is indexed
        [✓] parameter 1 is indexed
[✓] Approval(address,address,uint256) is present
        [✓] parameter 0 is indexed
```

```
        [] parameter 1 is indexed

        [] LatinaToken has increaseAllowance(address,uint256)
```

0x Guard