

Code Inspector

- Reports
- Assets
- Settings

← Reports

Fix potential vulnerabilities and bump solidity to... [View on GitHub](#)

Information

eonian-core

790fa1c

farm

#259

10 June 2024, 16:24

staging

Status

Completed

4m 53s (6 days ago)

Results

5 Low

22 Note

27 Total

Search for an issue... Filter by severity

L	`transfer` and `send` Calls Are No Longer Considered Best Practice	1 Instance	CONTRACT INSPECTOR	▼
When <code>transfer</code> or <code>send</code> calls are used to transfer ETH to an address, they forward a limited amount of gas. Given that gas...				
L	Unsafe ABI Encoding	1 Instance	CONTRACT INSPECTOR	▼
It is not an uncommon practice to use <code>abi.encodeWithSignature</code> or <code>abi.encodeWithSelector</code> to generate calldata for a lo...				
L	Floating Pragma	48 Instances	CONTRACT INSPECTOR	▼
Pragma directives should be fixed to clearly identify the Solidity version with which the contracts will be compiled.				
L	Missing Docstrings	154 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are multiple instances of code that does not have docstrings. For instance:				
L	Void Constructor Calls	3 Instances	CONTRACT INSPECTOR	▼
A parent constructor that is being called but is not present in a parent contract can indicate a flaw in the code, as the dev...				
N	Array Length Could Be Written to Stack	5 Instances	CONTRACT INSPECTOR	▼
In the EVM it is more gas-efficient to read values from stack than from memory or state. As in a for loop, where a value is r...				
N	External Calls Could Be Reentrancy Attack Vectors	2 Instances	CONTRACT INSPECTOR	▼
One of the major dangers of calling external contracts is that they can take over the control flow and take actions that the...				
N	`_disableInitializers()` Not Being Called From Initializable Contract Constructors	15 Instances	CONTRACT INSPECTOR	▼
An implementation contract in a proxy pattern allows anyone to call its <code>initialize</code> function. While not a direct security co...				
N	Incremental Update Is Not Wrapped in an Unchecked Block	7 Instances	CONTRACT INSPECTOR	▼
Since Solidity version 0.8.0 any arithmetic operations automatically check for over- and underflows which costs gas. Since...				
N	The Consistency of `msg.sender` or `msg.data` Usage Must Be Checked	11 Instances	CONTRACT INSPECTOR	▼
A contract may use the <code>_msgSender</code> and <code>_msgData</code> functions in certain cases where they allow meta transactions and have...				
N	Variables Are Initialized to Their Default Values	16 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, some variables are being initialized to their default values. For instance:				
N	Function Visibility Overly Permissive	119 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are various functions with unnecessarily permissive visibility. For instance:				
N	Todo Comments in the Code	3 Instances	CONTRACT INSPECTOR	▼
During development, having well described TODO/Fixme comments will make the process of tracking and solving them ea...				
N	Inconsistent Use of Named Returns	2 Instances	CONTRACT INSPECTOR	▼
To improve the readability of the contract, use the same return style in all of it is functions.				
N	Prefix Increment Operator `++i` Can Save Gas in Loops	8 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are multiple opportunities where this optimization could be applied. For instance:				
N	Lack of Indexed Event Parameters	14 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, several events do not have indexed parameters. For instance:				
N	Lack of Security Contact	47 Instances	CONTRACT INSPECTOR	▼
Providing a specific security contact (such as an email or ENS name) within a smart contract significantly simplifies the pr...				



<div><div>N</div><div>Possible Precision Loss Due to Division Before Multiplication</div></div>	3 Instances	CONTRACT INSPECTOR	▼
Solidity's integer division truncates. Thus, performing division before multiplication can lead to precision loss.			
<div><div>N</div><div>Functions Are Updating the State Without Event Emissions</div></div>	188 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, instances of functions that are updating the state without an event emission were found. For in...			
<div><div>N</div><div>Some Assignments Are Unnecessary</div></div>	4 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are unnecessary assignments of variables. For instance:			
<div><div>N</div><div>Unnecessary Casts</div></div>	7 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are unnecessary casts. For instance:			
<div><div>N</div><div>Unused Function Arguments</div></div>	1 Instance	CONTRACT INSPECTOR	▼
Within the Stub <code>ContractInvalidUpgrade</code> contract, the <code>needDisableInitializers</code> argument for the <code>constructor func...</code>			
<div><div>N</div><div>Unused Functions With Internal or Private Visibility</div></div>	7 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are unused functions. For instance:			
<div><div>N</div><div>Unused Imports</div></div>	24 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are imports that are unused and could be removed. For instance:			
<div><div>N</div><div>Unused Named Return Variables</div></div>	9 Instances	CONTRACT INSPECTOR	▼
Named return variables are a way to declare variables that are meant to be used within a function body for the purpose of...			
<div><div>N</div><div>Unused State Variables</div></div>	23 Instances	CONTRACT INSPECTOR	▼
Throughout the codebase, there are unused state variables. For instance:			
<div><div>N</div><div>Unused Struct</div></div>	1 Instance	CONTRACT INSPECTOR	▼
In <code>IRainMaker.sol</code> , the <code>CompMarketState</code> struct is unused.			

