Sri Lanka Institute of Information Technology

IT21299902 (ZAKEY M.S.M.A)

Smart Contract Competition

Eigen Layer Contest (Smart contract)

Web security – IE2062

B.Sc. (Hons) in Information Technology Specialization in cyber security.

Declaration:

- I hereby certify that no part of this assignment has been copied from any other work or any other source. No part of this assignment has been written/produced for me by another person.

- I hold a copy of this assignment that I can produce if the original is lost or damaged.

Project Details:

Case Study	Smart contract competition report
Date Of completion	04/05/2023

Table of contents.

Introduction.

Smart contracts are self-executing contracts that leverage blockchain technology to automate business processes and enable decentralized applications (dApps). They are written in programming languages like Solidity and deployed on blockchain platforms such as Ethereum. To use smart contracts, developers need to understand the coding concepts, syntax, and interactions with the blockchain network [1].

Benefits of smart contracts include transparency, efficiency, cost-effectiveness, and increased security. However, securing smart contracts is paramount due to potential vulnerabilities. Best practices include thorough testing, code audits, and adherence to coding standards. Smart contract competitions, such as the Eigen Layer Contest, are popular in the blockchain community, challenging participants to identify and exploit vulnerabilities in smart contracts to win prizes.

Eigen Layer is a well-known smart contract competition, and the author of the report plans to participate in it, showcasing the significance of such competitions in the field of blockchain technology. By understanding the fundamentals of smart contracts, their usage, benefits, security considerations, and participation in competitions like Eigen Layer, we can better appreciate the potential of blockchain technology and its impact on various industries [2].

1. Introduction to Smart contract auditing.

Smart contract auditing is the process of thoroughly reviewing and assessing a smart contract's code to identify and correct any potential security flaws, errors, or other issues that might compromise the smart contract's functionality or endanger investors' funds. The code is examined during the auditing process using a range of techniques and tools, including manual inspection, automated testing, and vulnerability scanners [3].

The goal is to ensure the smart contract's security, reliability, and correct operation. After the audit, the auditors provide a comprehensive report outlining their findings and recommendations for improving the contract's security and usability [4]. A smart contract is required for any blockchain-based project that wishes to ensure that its contracts are trustworthy and safe. Also, there are different phases of smart contracts auditing [5].

i. **Requirement Analysis**
ii. **Code Review**
iii. **Security Assessment**
iv. **Functionality Testing**
v. **Gas Optimization**
vi. **Documentation Review**
vii. **Report and Recommendations**

2. Smart contract vulnerabilities.

i. Reentrancy attack.

Attackers can call a function more than once before the call before it has concluded thanks to reentrancy. Unexpected and harmful outcomes like money theft or unauthorized access to data might occur from this.

There are three sorts of reentrancy attacks: several methods for the same contract; various techniques for other contracts; and the same approach for the same contract [6].

ii. Integer overflow and underflow.

An integer overflow occurs when the result of a mathematical operation exceeds the maximum value that can be stored in the variable; an integer underflow occurs when the result is less than the minimum value that may be stored. These weaknesses can be used by attackers to cause applications to act unexpectedly and perhaps destructively [7].

iii. Denial of service attacks (DOS)

Contracts are vulnerable to denial-of-service (DoS) attacks, in which an attacker purposefully uses a large amount of resources, rendering the contract unavailable or triggering disturbances in the whole blockchain network. Techniques like infinite loops, excessive processing, or resource exhaustion can be used to achieve this [8].

iv. Block gas limit

The Ethereum network has a block gas limit that prevents blocks from getting out of control in size. It simply refers to the maximum amount of gas that transactions in a block can use. On the other hand, if a transaction uses too much gas, it won't fit in a block and won't be carried out [9].

v. Front running.

In a front-running assault, a hostile actor takes use of their knowledge of impending transactions to obtain an unfair advantage in blockchain-based systems like Ethereum.

In a front-running assault, a party, usually a miner or trader, inserts their own transaction into the blockchain before that of another user to profit from the price fluctuations brought on by the second transaction [10].

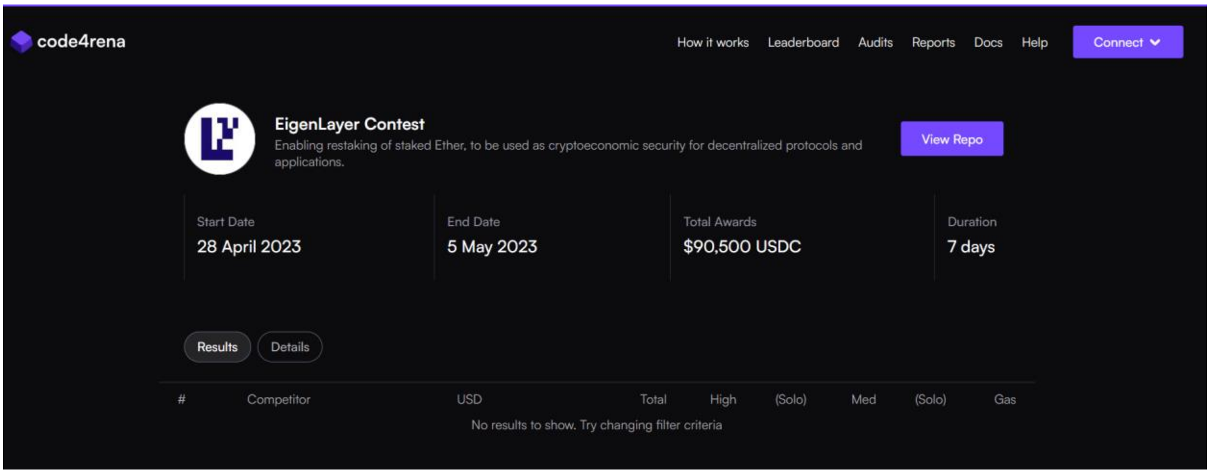
3. Introduction to the Eigen Layer Contest.

Eigen Layer is a popular smart contract competition that challenges participants to test their skills in identifying and exploiting vulnerabilities in smart contracts. The competition is typically structured into multiple levels, each with increasing difficulty.

Participants are required to analyze and interact with smart contracts written in Solidity, identify potential security flaws, and craft exploits to breach the contract's defenses. The competition may involve tasks such as reverse engineering, code analysis, and exploit development to gain unauthorized access to contract functions or manipulate contract state.

Players may need to use various tools, techniques, and knowledge of blockchain technology to successfully complete the challenges. Playing Eigen Layer contest requires a strong understanding of smart contracts, Solidity programming, and security best practices, making it an engaging and challenging experience for participants looking to test their skills in the field of blockchain security.

i. Proof to smart contract competition.



EigenLayer Contest

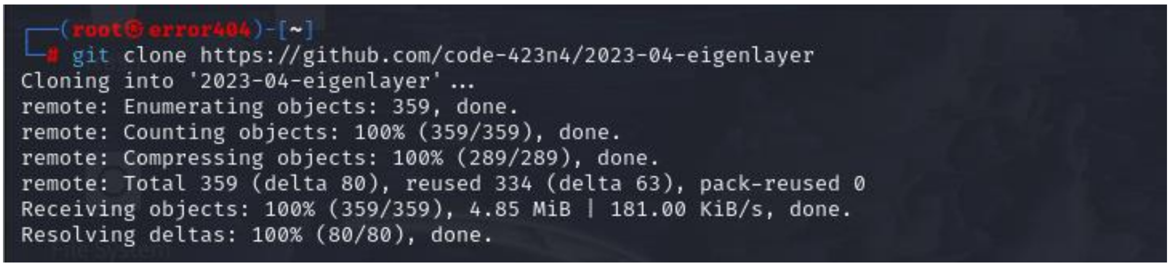
Enabling restaking of staked Ether, to be used as cryptoeconomic security for decentralized protocols and applications.

Start Date	End Date	Total Awards	Duration
28 April 2023	5 May 2023	$90,500 USDC	7 days

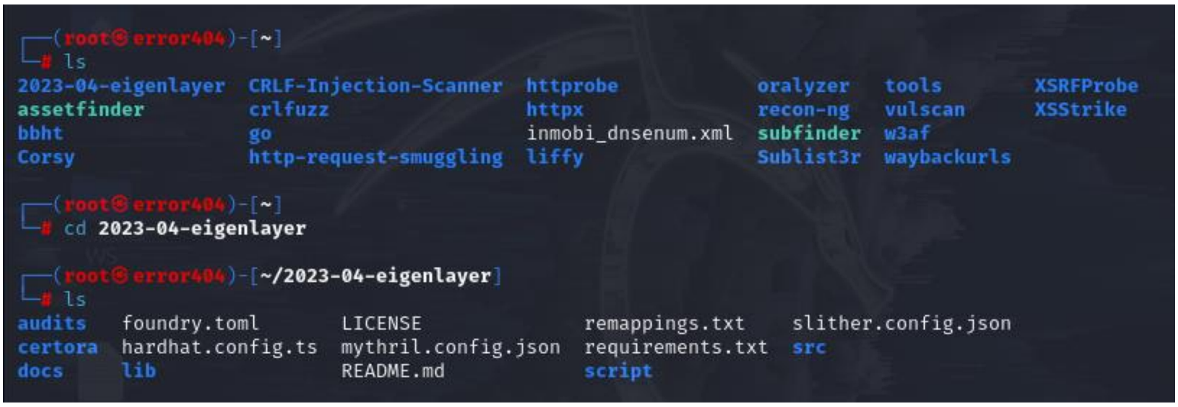
#	Competitor	USD	Total	High	(Solo)	Med	(Solo)	Gas

No results to show. Try changing filter criteria

ii. Cloning the smart contract repository to my Linux environment.

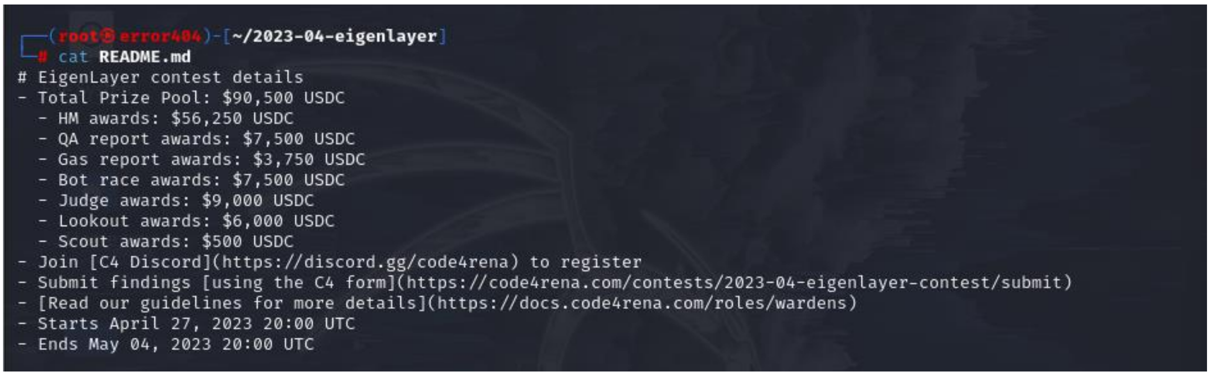


```
┌──(root㉿error404)-[~]
└─# git clone https://github.com/code-423n4/2023-04-eigenlayer
Cloning into '2023-04-eigenlayer'...
remote: Enumerating objects: 359, done.
remote: Counting objects: 100% (359/359), done.
remote: Compressing objects: 100% (289/289), done.
remote: Total 359 (delta 80), reused 334 (delta 63), pack-reused 0
Receiving objects: 100% (359/359), 4.85 MiB | 181.00 KiB/s, done.
Resolving deltas: 100% (80/80), done.
```

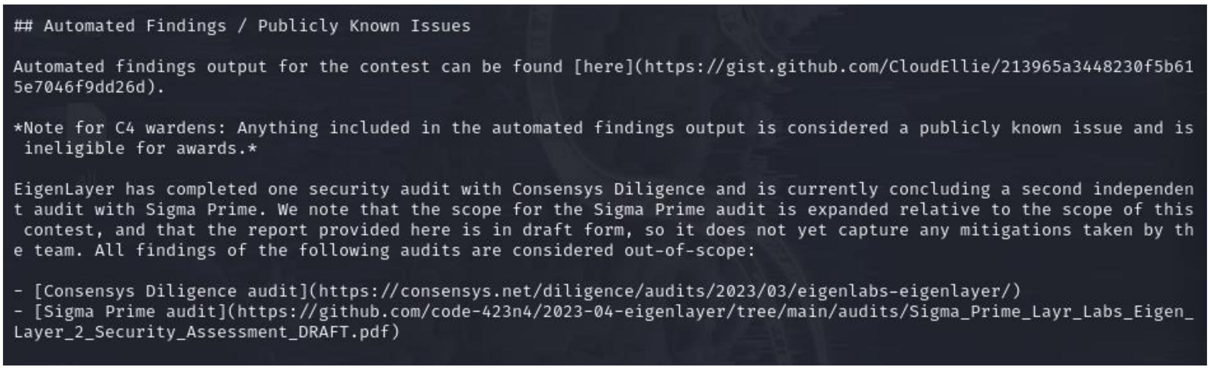


```
┌──(root㉿error404)-[~]
└─# ls
2023-04-eigenlayer  CRLF-Injection-Scanner  httprobe           oralyzer   tools        XSRFProbe
assetfinder         crlfuzz                 httpx              recon-ng   vulscan      XSStrike
bbht                go                      inmobi_dnsenum.xml subfinder  w3af
Corsy               http-request-smuggling  liffy              Sublist3r  waybackurls

┌──(root㉿error404)-[~]
└─# cd 2023-04-eigenlayer

┌──(root㉿error404)-[~/2023-04-eigenlayer]
└─# ls
audits   foundry.toml      LICENSE              remappings.txt    slither.config.json
certora  hardhat.config.ts mythril.config.json  requirements.txt  src
docs     lib               README.md            script
```

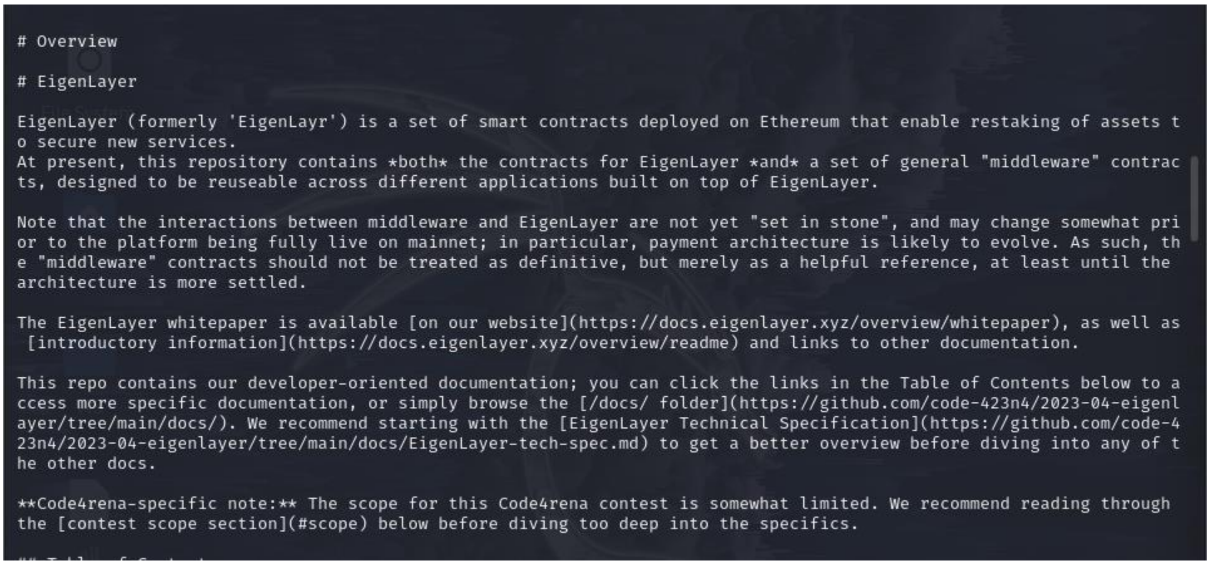
iii. Eigen Layer competition guidelines.



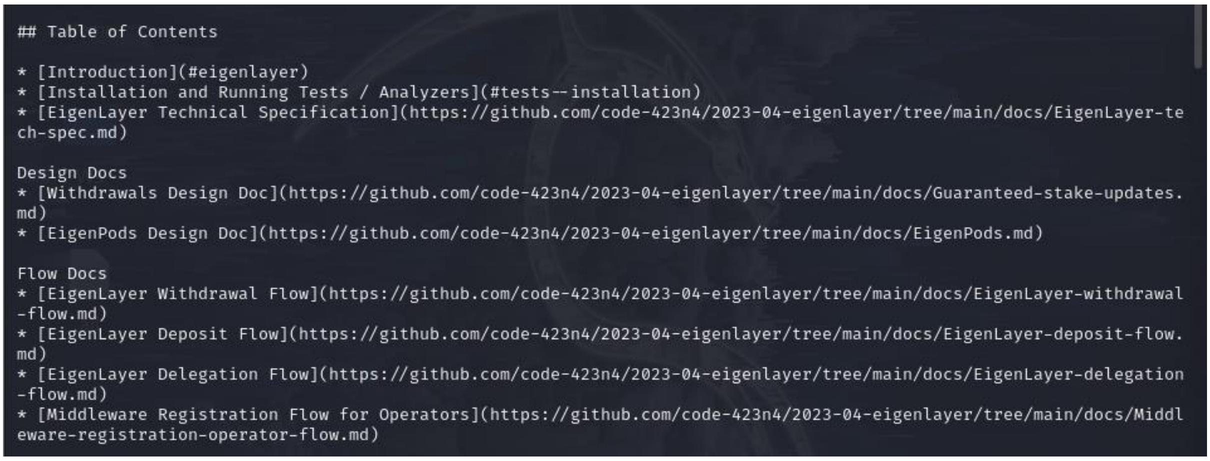
```
┌──(root㉿error404)-[~/2023-04-eigenlayer]
└─# cat README.md
# EigenLayer contest details
- Total Prize Pool: $90,500 USDC
  - HM awards: $56,250 USDC
  - QA report awards: $7,500 USDC
  - Gas report awards: $3,750 USDC
  - Bot race awards: $7,500 USDC
  - Judge awards: $9,000 USDC
  - Lookout awards: $6,000 USDC
  - Scout awards: $500 USDC
- Join [C4 Discord](https://discord.gg/code4rena) to register
- Submit findings [using the C4 form](https://code4rena.com/contests/2023-04-eigenlayer-contest/submit)
- [Read our guidelines for more details](https://docs.code4rena.com/roles/wardens)
- Starts April 27, 2023 20:00 UTC
- Ends May 04, 2023 20:00 UTC
```



```
## Automated Findings / Publicly Known Issues

Automated findings output for the contest can be found [here](https://gist.github.com/CloudEllie/213965a3448230f5b61
5e7046f9dd26d).

*Note for C4 wardens: Anything included in the automated findings output is considered a publicly known issue and is
 ineligible for awards.*

EigenLayer has completed one security audit with Consensys Diligence and is currently concluding a second independen
t audit with Sigma Prime. We note that the scope for the Sigma Prime audit is expanded relative to the scope of this
 contest, and that the report provided here is in draft form, so it does not yet capture any mitigations taken by th
e team. All findings of the following audits are considered out-of-scope:

- [Consensys Diligence audit](https://consensys.net/diligence/audits/2023/03/eigenlabs-eigenlayer/)
- [Sigma Prime audit](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/audits/Sigma_Prime_Layr_Labs_Eigen_
Layer_2_Security_Assessment_DRAFT.pdf)
```

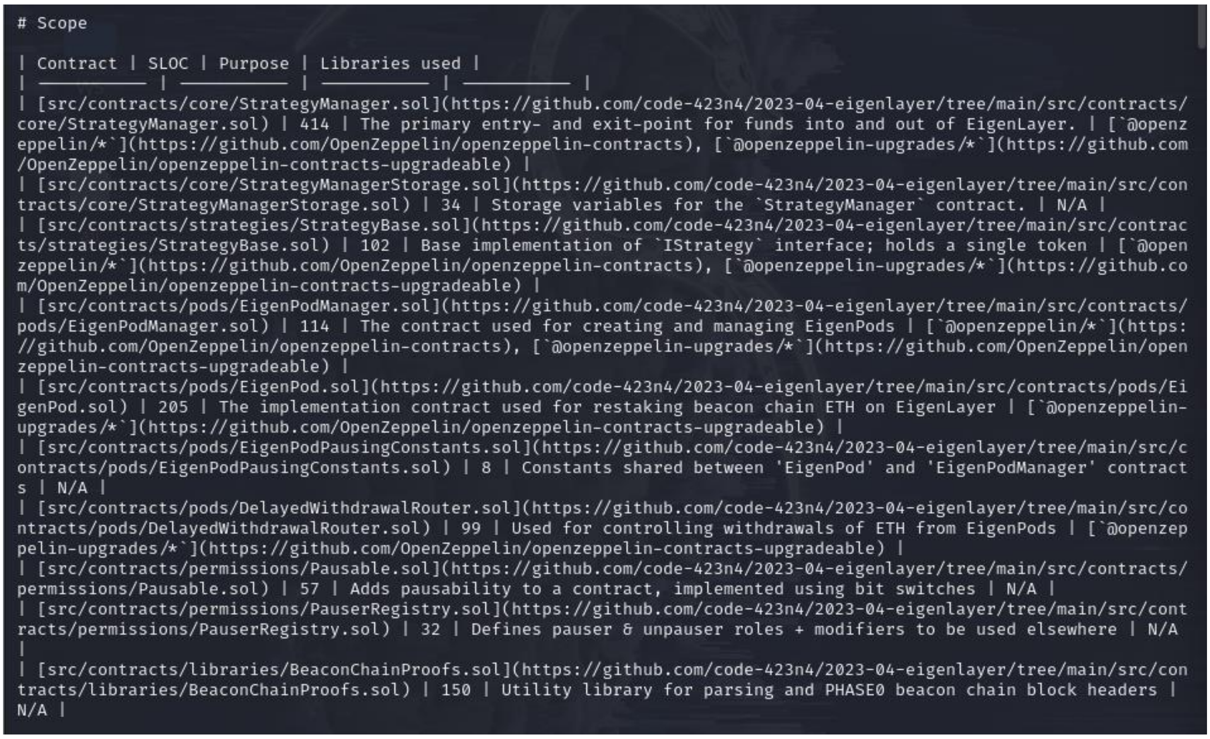


```
# Overview

# EigenLayer

EigenLayer (formerly 'EigenLayr') is a set of smart contracts deployed on Ethereum that enable restaking of assets t
o secure new services.
At present, this repository contains *both* the contracts for EigenLayer *and* a set of general "middleware" contrac
ts, designed to be reuseable across different applications built on top of EigenLayer.

Note that the interactions between middleware and EigenLayer are not yet "set in stone", and may change somewhat pri
or to the platform being fully live on mainnet; in particular, payment architecture is likely to evolve. As such, th
e "middleware" contracts should not be treated as definitive, but merely as a helpful reference, at least until the
architecture is more settled.

The EigenLayer whitepaper is available [on our website](https://docs.eigenlayer.xyz/overview/whitepaper), as well as
 [introductory information](https://docs.eigenlayer.xyz/overview/readme) and links to other documentation.

This repo contains our developer-oriented documentation; you can click the links in the Table of Contents below to a
ccess more specific documentation, or simply browse the [/docs/ folder](https://github.com/code-423n4/2023-04-eigenl
ayer/tree/main/docs/). We recommend starting with the [EigenLayer Technical Specification](https://github.com/code-4
23n4/2023-04-eigenlayer/tree/main/docs/EigenLayer-tech-spec.md) to get a better overview before diving into any of t
he other docs.

**Code4rena-specific note:** The scope for this Code4rena contest is somewhat limited. We recommend reading through
the [contest scope section](#scope) below before diving too deep into the specifics.
```

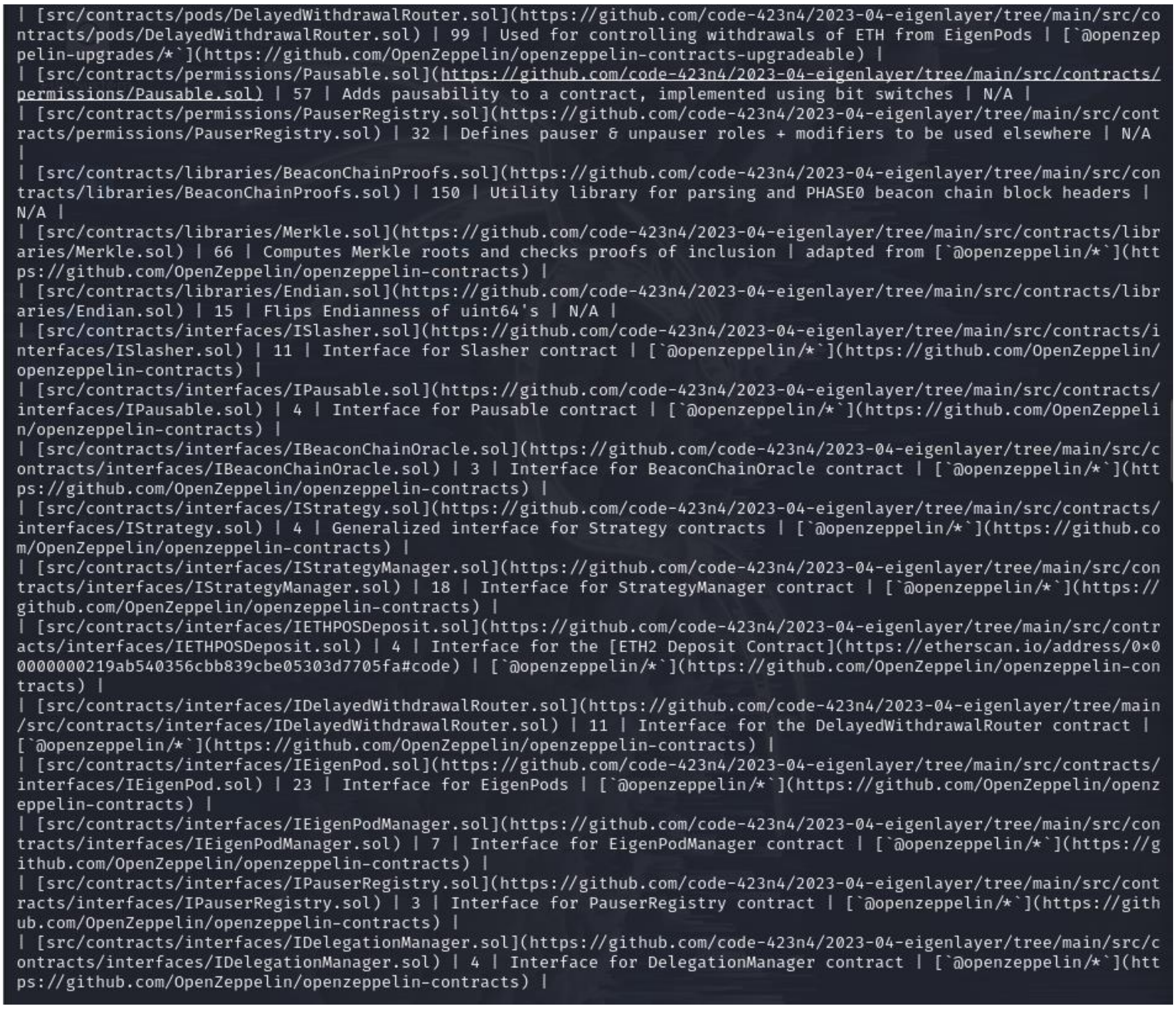


```
## Table of Contents

* [Introduction](#eigenlayer)
* [Installation and Running Tests / Analyzers](#tests--installation)
* [EigenLayer Technical Specification](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/EigenLayer-te
ch-spec.md)

Design Docs
* [Withdrawals Design Doc](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/Guaranteed-stake-updates.
md)
* [EigenPods Design Doc](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/EigenPods.md)

Flow Docs
* [EigenLayer Withdrawal Flow](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/EigenLayer-withdrawal
-flow.md)
* [EigenLayer Deposit Flow](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/EigenLayer-deposit-flow.
md)
* [EigenLayer Delegation Flow](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/EigenLayer-delegation
-flow.md)
* [Middleware Registration Flow for Operators](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/docs/Middl
eware-registration-operator-flow.md)
```

iv. In scope sol files (smart contracts).

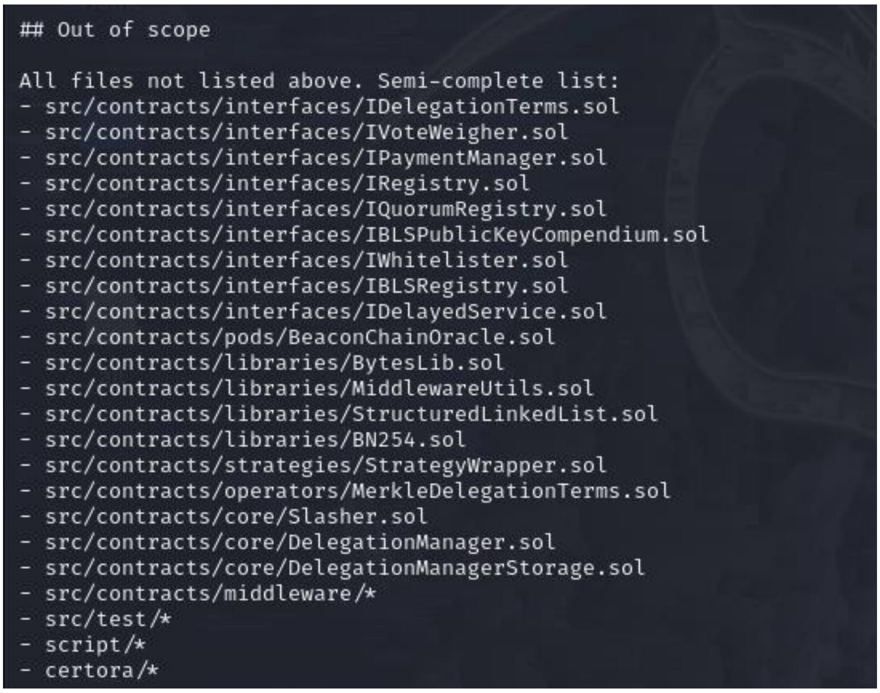


```
# Scope

| Contract | SLOC | Purpose | Libraries used |
| ----------- | ----------- | ----------- | ----------- |
| [src/contracts/core/StrategyManager.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
core/StrategyManager.sol) | 414 | The primary entry- and exit-point for funds into and out of EigenLayer. | [`@openz
eppelin/*`](https://github.com/OpenZeppelin/openzeppelin-contracts), [`@openzeppelin-upgrades/*`](https://github.com
/OpenZeppelin/openzeppelin-contracts-upgradeable) |
| [src/contracts/core/StrategyManagerStorage.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/con
tracts/core/StrategyManagerStorage.sol) | 34 | Storage variables for the `StrategyManager` contract. | N/A |
| [src/contracts/strategies/StrategyBase.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contrac
ts/strategies/StrategyBase.sol) | 102 | Base implementation of `IStrategy` interface; holds a single token | [`@open
zeppelin/*`](https://github.com/OpenZeppelin/openzeppelin-contracts), [`@openzeppelin-upgrades/*`](https://github.co
m/OpenZeppelin/openzeppelin-contracts-upgradeable) |
| [src/contracts/pods/EigenPodManager.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
pods/EigenPodManager.sol) | 114 | The contract used for creating and managing EigenPods | [`@openzeppelin/*`](https:
//github.com/OpenZeppelin/openzeppelin-contracts), [`@openzeppelin-upgrades/*`](https://github.com/OpenZeppelin/open
zeppelin-contracts-upgradeable) |
| [src/contracts/pods/EigenPod.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/pods/Ei
genPod.sol) | 205 | The implementation contract used for restaking beacon chain ETH on EigenLayer | [`@openzeppelin-
upgrades/*`](https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable) |
| [src/contracts/pods/EigenPodPausingConstants.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/c
ontracts/pods/EigenPodPausingConstants.sol) | 8 | Constants shared between 'EigenPod' and 'EigenPodManager' contract
s | N/A |
| [src/contracts/pods/DelayedWithdrawalRouter.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/co
ntracts/pods/DelayedWithdrawalRouter.sol) | 99 | Used for controlling withdrawals of ETH from EigenPods | [`@openzep
pelin-upgrades/*`](https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable) |
| [src/contracts/permissions/Pausable.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
permissions/Pausable.sol) | 57 | Adds pausability to a contract, implemented using bit switches | N/A |
| [src/contracts/permissions/PauserRegistry.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/cont
racts/permissions/PauserRegistry.sol) | 32 | Defines pauser & unpauser roles + modifiers to be used elsewhere | N/A
|
| [src/contracts/libraries/BeaconChainProofs.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/con
tracts/libraries/BeaconChainProofs.sol) | 150 | Utility library for parsing and PHASE0 beacon chain block headers |
N/A |
```



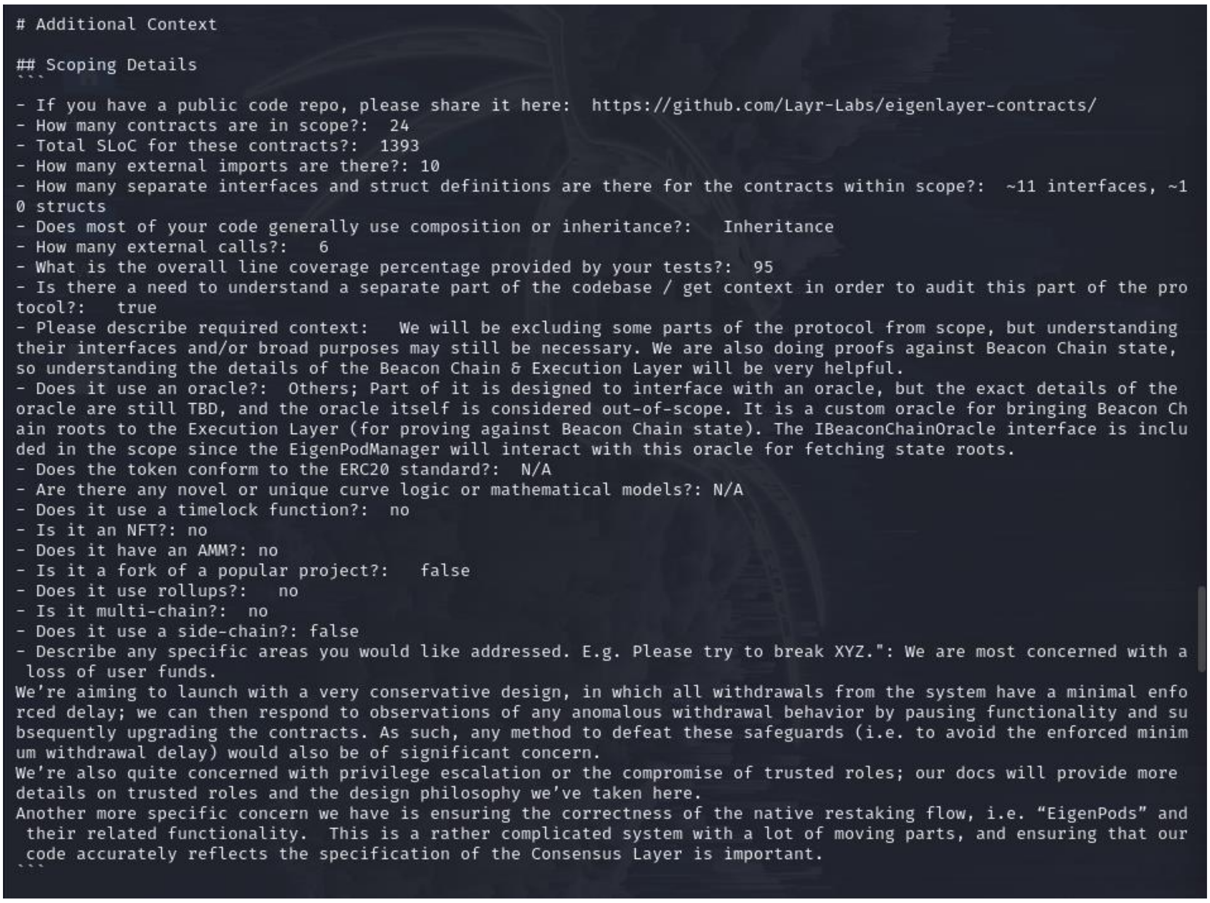
```
| [src/contracts/pods/DelayedWithdrawalRouter.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/co
ntracts/pods/DelayedWithdrawalRouter.sol) | 99 | Used for controlling withdrawals of ETH from EigenPods | [`@openzep
pelin-upgrades/*`](https://github.com/OpenZeppelin/openzeppelin-contracts-upgradeable) |
| [src/contracts/permissions/Pausable.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
permissions/Pausable.sol) | 57 | Adds pausability to a contract, implemented using bit switches | N/A |
| [src/contracts/permissions/PauserRegistry.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/cont
racts/permissions/PauserRegistry.sol) | 32 | Defines pauser & unpauser roles + modifiers to be used elsewhere | N/A
|
| [src/contracts/libraries/BeaconChainProofs.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/con
tracts/libraries/BeaconChainProofs.sol) | 150 | Utility library for parsing and PHASE0 beacon chain block headers |
N/A |
| [src/contracts/libraries/Merkle.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/libr
aries/Merkle.sol) | 66 | Computes Merkle roots and checks proofs of inclusion | adapted from [`@openzeppelin/*`](htt
ps://github.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/libraries/Endian.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/libr
aries/Endian.sol) | 15 | Flips Endianness of uint64's | N/A |
| [src/contracts/interfaces/ISlasher.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/i
nterfaces/ISlasher.sol) | 11 | Interface for Slasher contract | [`@openzeppelin/*`](https://github.com/OpenZeppelin/
openzeppelin-contracts) |
| [src/contracts/interfaces/IPausable.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
interfaces/IPausable.sol) | 4 | Interface for Pausable contract | [`@openzeppelin/*`](https://github.com/OpenZeppeli
n/openzeppelin-contracts) |
| [src/contracts/interfaces/IBeaconChainOracle.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/c
ontracts/interfaces/IBeaconChainOracle.sol) | 3 | Interface for BeaconChainOracle contract | [`@openzeppelin/*`](htt
ps://github.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IStrategy.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
interfaces/IStrategy.sol) | 4 | Generalized interface for Strategy contracts | [`@openzeppelin/*`](https://github.co
m/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IStrategyManager.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/con
tracts/interfaces/IStrategyManager.sol) | 18 | Interface for StrategyManager contract | [`@openzeppelin/*`](https://
github.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IETHPOSDeposit.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contr
acts/interfaces/IETHPOSDeposit.sol) | 4 | Interface for the [ETH2 Deposit Contract](https://etherscan.io/address/0×0
0000000219ab540356cbb839cbe05303d7705fa#code) | [`@openzeppelin/*`](https://github.com/OpenZeppelin/openzeppelin-con
tracts) |
| [src/contracts/interfaces/IDelayedWithdrawalRouter.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main
/src/contracts/interfaces/IDelayedWithdrawalRouter.sol) | 11 | Interface for the DelayedWithdrawalRouter contract |
[`@openzeppelin/*`](https://github.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IEigenPod.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/contracts/
interfaces/IEigenPod.sol) | 23 | Interface for EigenPods | [`@openzeppelin/*`](https://github.com/OpenZeppelin/openz
eppelin-contracts) |
| [src/contracts/interfaces/IEigenPodManager.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/con
tracts/interfaces/IEigenPodManager.sol) | 7 | Interface for EigenPodManager contract | [`@openzeppelin/*`](https://g
ithub.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IPauserRegistry.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/cont
racts/interfaces/IPauserRegistry.sol) | 3 | Interface for PauserRegistry contract | [`@openzeppelin/*`](https://gith
ub.com/OpenZeppelin/openzeppelin-contracts) |
| [src/contracts/interfaces/IDelegationManager.sol](https://github.com/code-423n4/2023-04-eigenlayer/tree/main/src/c
ontracts/interfaces/IDelegationManager.sol) | 4 | Interface for DelegationManager contract | [`@openzeppelin/*`](htt
ps://github.com/OpenZeppelin/openzeppelin-contracts) |
```

v. Out of scope sol files (smart contracts).



```
## Out of scope

All files not listed above. Semi-complete list:
- src/contracts/interfaces/IDelegationTerms.sol
- src/contracts/interfaces/IVoteWeigher.sol
- src/contracts/interfaces/IPaymentManager.sol
- src/contracts/interfaces/IRegistry.sol
- src/contracts/interfaces/IQuorumRegistry.sol
- src/contracts/interfaces/IBLSPublicKeyCompendium.sol
- src/contracts/interfaces/IWhitelister.sol
- src/contracts/interfaces/IBLSRegistry.sol
- src/contracts/interfaces/IDelayedService.sol
- src/contracts/pods/BeaconChainOracle.sol
- src/contracts/libraries/BytesLib.sol
- src/contracts/libraries/MiddlewareUtils.sol
- src/contracts/libraries/StructuredLinkedList.sol
- src/contracts/libraries/BN254.sol
- src/contracts/strategies/StrategyWrapper.sol
- src/contracts/operators/MerkleDelegationTerms.sol
- src/contracts/core/Slasher.sol
- src/contracts/core/DelegationManager.sol
- src/contracts/core/DelegationManagerStorage.sol
- src/contracts/middleware/*
- src/test/*
- script/*
- certora/*
```

vi. Additional details about the smart contract competition.

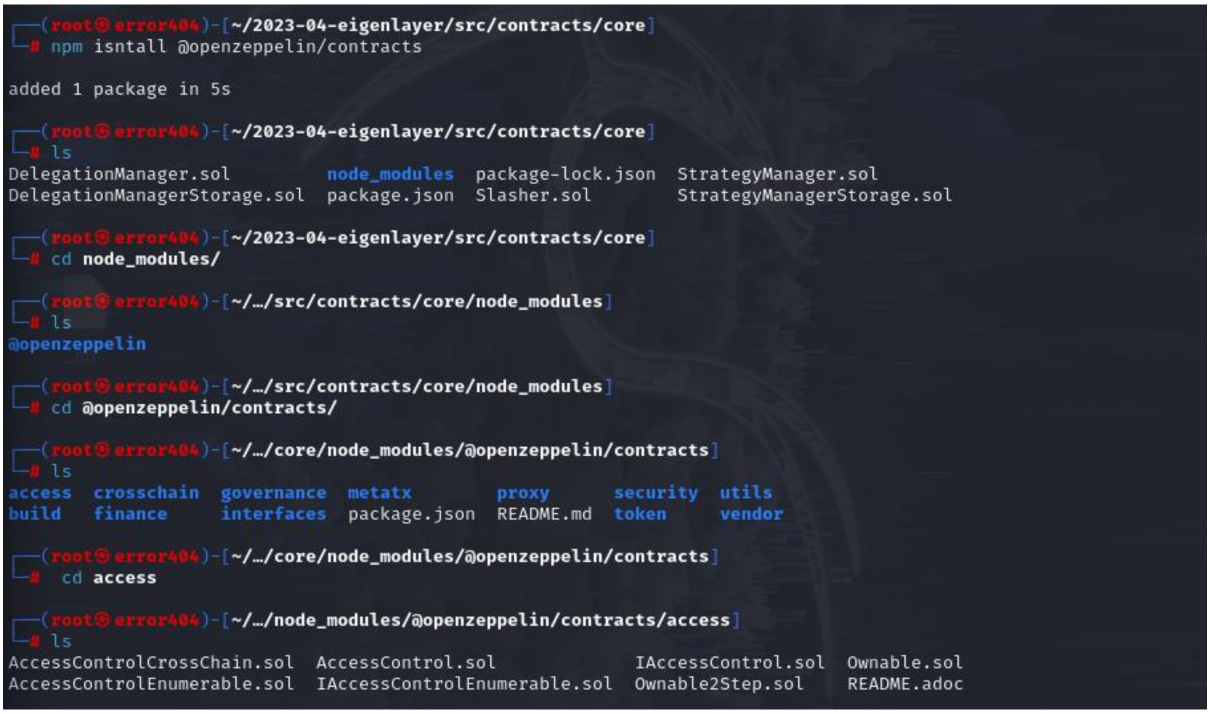


~~~
# Additional Context

## Scoping Details
```
- If you have a public code repo, please share it here:  https://github.com/Layr-Labs/eigenlayer-contracts/
- How many contracts are in scope?:  24
- Total SLoC for these contracts?:  1393
- How many external imports are there?: 10
- How many separate interfaces and struct definitions are there for the contracts within scope?:  ~11 interfaces, ~1
0 structs
- Does most of your code generally use composition or inheritance?:   Inheritance
- How many external calls?:   6
- What is the overall line coverage percentage provided by your tests?:  95
- Is there a need to understand a separate part of the codebase / get context in order to audit this part of the pro
tocol?:   true
- Please describe required context:   We will be excluding some parts of the protocol from scope, but understanding
their interfaces and/or broad purposes may still be necessary. We are also doing proofs against Beacon Chain state,
so understanding the details of the Beacon Chain & Execution Layer will be very helpful.
- Does it use an oracle?:  Others; Part of it is designed to interface with an oracle, but the exact details of the
oracle are still TBD, and the oracle itself is considered out-of-scope. It is a custom oracle for bringing Beacon Ch
ain roots to the Execution Layer (for proving against Beacon Chain state). The IBeaconChainOracle interface is inclu
ded in the scope since the EigenPodManager will interact with this oracle for fetching state roots.
- Does the token conform to the ERC20 standard?:  N/A
- Are there any novel or unique curve logic or mathematical models?: N/A
- Does it use a timelock function?:  no
- Is it an NFT?: no
- Does it have an AMM?: no
- Is it a fork of a popular project?:   false
- Does it use rollups?:   no
- Is it multi-chain?:  no
- Does it use a side-chain?: false
- Describe any specific areas you would like addressed. E.g. Please try to break XYZ.": We are most concerned with a
 loss of user funds.
We're aiming to launch with a very conservative design, in which all withdrawals from the system have a minimal enfo
rced delay; we can then respond to observations of any anomalous withdrawal behavior by pausing functionality and su
bsequently upgrading the contracts. As such, any method to defeat these safeguards (i.e. to avoid the enforced minim
um withdrawal delay) would also be of significant concern.
We're also quite concerned with privilege escalation or the compromise of trusted roles; our docs will provide more
details on trusted roles and the design philosophy we've taken here.
Another more specific concern we have is ensuring the correctness of the native restaking flow, i.e. "EigenPods" and
 their related functionality.  This is a rather complicated system with a lot of moving parts, and ensuring that our
 code accurately reflects the specification of the Consensus Layer is important.
```
~~~

4. Downloading relevant sol files and tools.

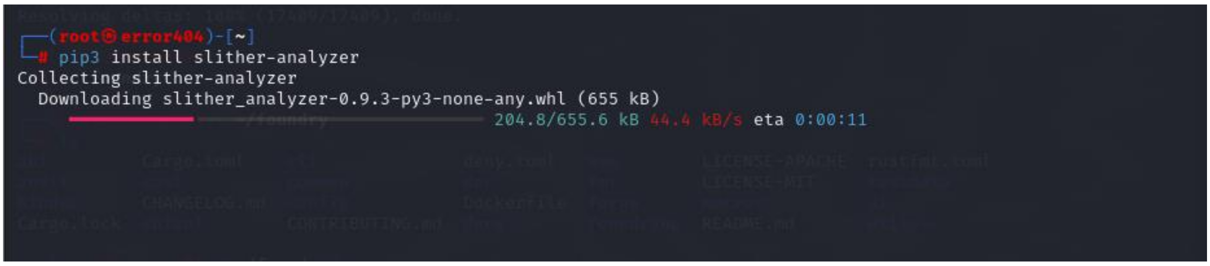
i. Installing open zeppelin project files.

After doing above mentioned steps are needed to install contracts of the eigen layer contract from open zeppelin. To download the contract, I have to use the command npm install @openzeppelin/contracts in respective directory.

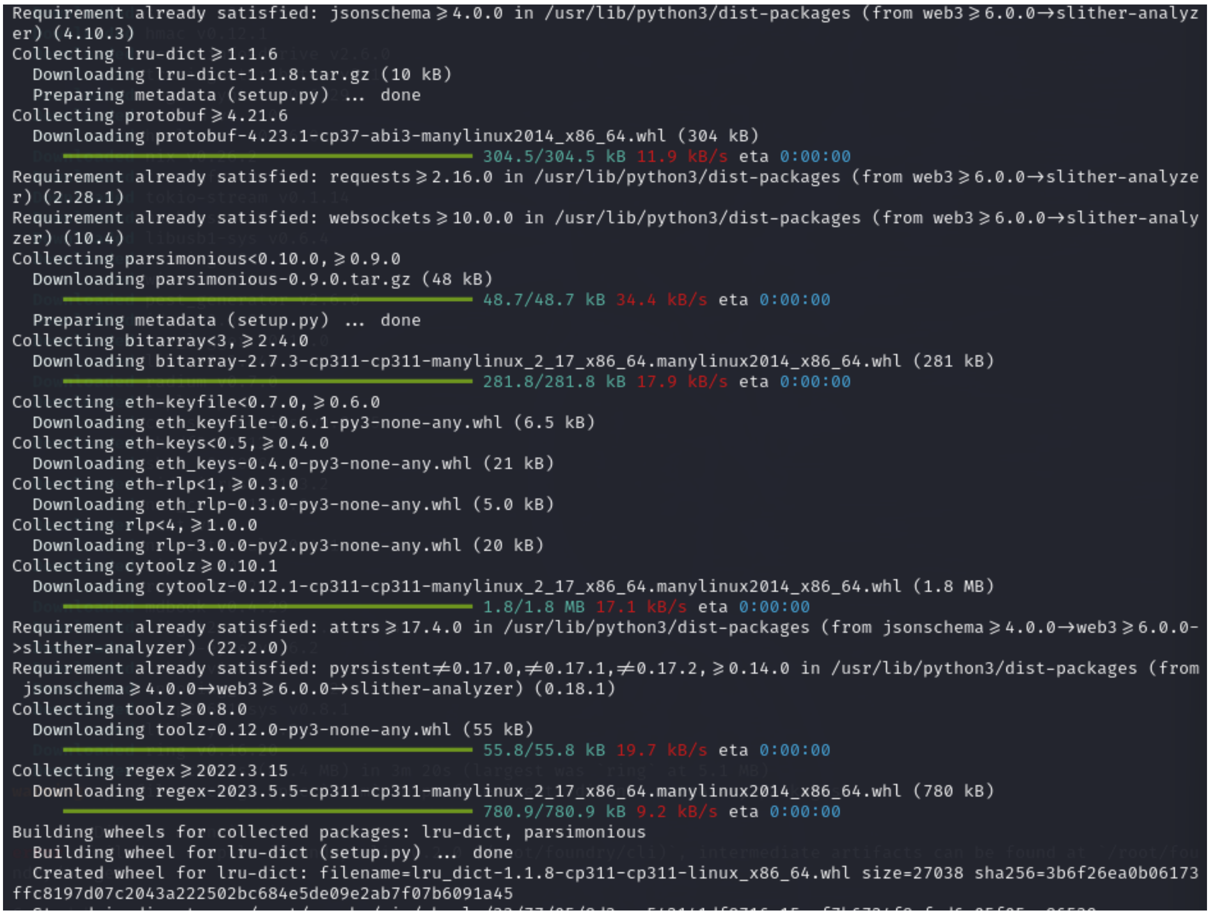


```
┌──(root㉿error404)-[~/2023-04-eigenlayer/src/contracts/core]
└─# npm isntall @openzeppelin/contracts

added 1 package in 5s

┌──(root㉿error404)-[~/2023-04-eigenlayer/src/contracts/core]
└─# ls
DelegationManager.sol         node_modules  package-lock.json  StrategyManager.sol
DelegationManagerStorage.sol  package.json  Slasher.sol        StrategyManagerStorage.sol

┌──(root㉿error404)-[~/2023-04-eigenlayer/src/contracts/core]
└─# cd node_modules/

┌──(root㉿error404)-[~/…/src/contracts/core/node_modules]
└─# ls
@openzeppelin

┌──(root㉿error404)-[~/…/src/contracts/core/node_modules]
└─# cd @openzeppelin/contracts/

┌──(root㉿error404)-[~/…/core/node_modules/@openzeppelin/contracts]
└─# ls
access  crosschain  governance  metatx        proxy      security  utils
build   finance     interfaces  package.json  README.md  token     vendor

┌──(root㉿error404)-[~/…/core/node_modules/@openzeppelin/contracts]
└─#  cd access

┌──(root㉿error404)-[~/…/node_modules/@openzeppelin/contracts/access]
└─# ls
AccessControlCrossChain.sol   AccessControl.sol            IAccessControl.sol  Ownable.sol
AccessControlEnumerable.sol   IAccessControlEnumerable.sol Ownable2Step.sol    README.adoc
```

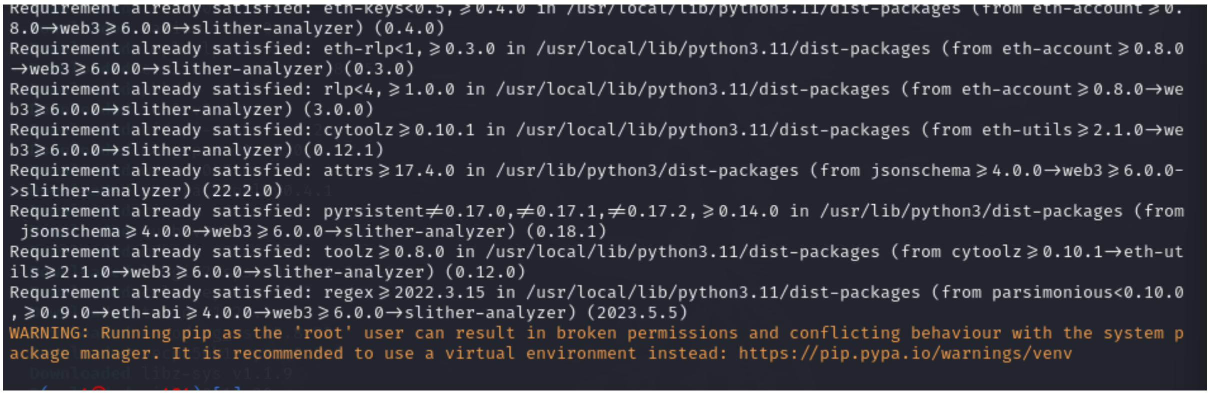
ii. Installing Slither automated smart contract analyzer.



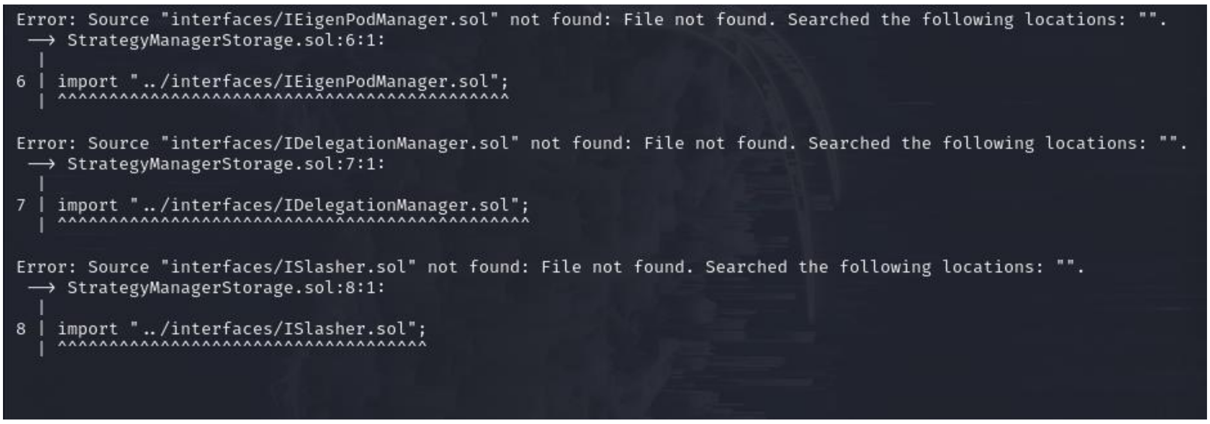
```
┌──(root㉿error404)-[~]
└─# pip3 install slither-analyzer
Collecting slither-analyzer
  Downloading slither_analyzer-0.9.3-py3-none-any.whl (655 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 204.8/655.6 kB 44.4 kB/s eta 0:00:11
```



```
Requirement already satisfied: jsonschema≥4.0.0 in /usr/lib/python3/dist-packages (from web3≥6.0.0→slither-analyz
er) (4.10.3)
Collecting lru-dict≥1.1.6
  Downloading lru-dict-1.1.8.tar.gz (10 kB)
  Preparing metadata (setup.py) ... done
Collecting protobuf≥4.21.6
  Downloading protobuf-4.23.1-cp37-abi3-manylinux2014_x86_64.whl (304 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 304.5/304.5 kB 11.9 kB/s eta 0:00:00
Requirement already satisfied: requests≥2.16.0 in /usr/lib/python3/dist-packages (from web3≥6.0.0→slither-analyze
r) (2.28.1)
Requirement already satisfied: websockets≥10.0.0 in /usr/lib/python3/dist-packages (from web3≥6.0.0→slither-analy
zer) (10.4)
Collecting parsimonious<0.10.0,≥0.9.0
  Downloading parsimonious-0.9.0.tar.gz (48 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.7/48.7 kB 34.4 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting bitarray<3,≥2.4.0
  Downloading bitarray-2.7.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (281 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.8/281.8 kB 17.9 kB/s eta 0:00:00
Collecting eth-keyfile<0.7.0,≥0.6.0
  Downloading eth_keyfile-0.6.1-py3-none-any.whl (6.5 kB)
Collecting eth-keys<0.5,≥0.4.0
  Downloading eth_keys-0.4.0-py3-none-any.whl (21 kB)
Collecting eth-rlp<1,≥0.3.0
  Downloading eth_rlp-0.3.0-py3-none-any.whl (5.0 kB)
Collecting rlp<4,≥1.0.0
  Downloading rlp-3.0.0-py2.py3-none-any.whl (20 kB)
Collecting cytoolz≥0.10.1
  Downloading cytoolz-0.12.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 17.1 kB/s eta 0:00:00
Requirement already satisfied: attrs≥17.4.0 in /usr/lib/python3/dist-packages (from jsonschema≥4.0.0→web3≥6.0.0-
>slither-analyzer) (22.2.0)
Requirement already satisfied: pyrsistent≠0.17.0,≠0.17.1,≠0.17.2,≥0.14.0 in /usr/lib/python3/dist-packages (from
 jsonschema≥4.0.0→web3≥6.0.0→slither-analyzer) (0.18.1)
Collecting toolz≥0.8.0
  Downloading toolz-0.12.0-py3-none-any.whl (55 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.8/55.8 kB 19.7 kB/s eta 0:00:00
Collecting regex≥2022.3.15
  Downloading regex-2023.5.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (780 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 780.9/780.9 kB 9.2 kB/s eta 0:00:00
Building wheels for collected packages: lru-dict, parsimonious
  Building wheel for lru-dict (setup.py) ... done
  Created wheel for lru-dict: filename=lru_dict-1.1.8-cp311-cp311-linux_x86_64.whl size=27038 sha256=3b6f26ea0b06173
ffc8197d07c2043a222502bc684e5de09e2ab7f07b6091a45
```



```
Requirement already satisfied: eth-keys<0.5,≥0.4.0 in /usr/local/lib/python3.11/dist-packages (from eth-account≥0.
8.0→web3≥6.0.0→slither-analyzer) (0.4.0)
Requirement already satisfied: eth-rlp<1,≥0.3.0 in /usr/local/lib/python3.11/dist-packages (from eth-account≥0.8.0
→web3≥6.0.0→slither-analyzer) (0.3.0)
Requirement already satisfied: rlp<4,≥1.0.0 in /usr/local/lib/python3.11/dist-packages (from eth-account≥0.8.0→we
b3≥6.0.0→slither-analyzer) (3.0.0)
Requirement already satisfied: cytoolz≥0.10.1 in /usr/local/lib/python3.11/dist-packages (from eth-utils≥2.1.0→we
b3≥6.0.0→slither-analyzer) (0.12.1)
Requirement already satisfied: attrs≥17.4.0 in /usr/lib/python3/dist-packages (from jsonschema≥4.0.0→web3≥6.0.0-
>slither-analyzer) (22.2.0)
Requirement already satisfied: pyrsistent≠0.17.0,≠0.17.1,≠0.17.2,≥0.14.0 in /usr/lib/python3/dist-packages (from
 jsonschema≥4.0.0→web3≥6.0.0→slither-analyzer) (0.18.1)
Requirement already satisfied: toolz≥0.8.0 in /usr/local/lib/python3.11/dist-packages (from cytoolz≥0.10.1→eth-ut
ils≥2.1.0→web3≥6.0.0→slither-analyzer) (0.12.0)
Requirement already satisfied: regex≥2022.3.15 in /usr/local/lib/python3.11/dist-packages (from parsimonious<0.10.0
,≥0.9.0→eth-abi≥4.0.0→web3≥6.0.0→slither-analyzer) (2023.5.5)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

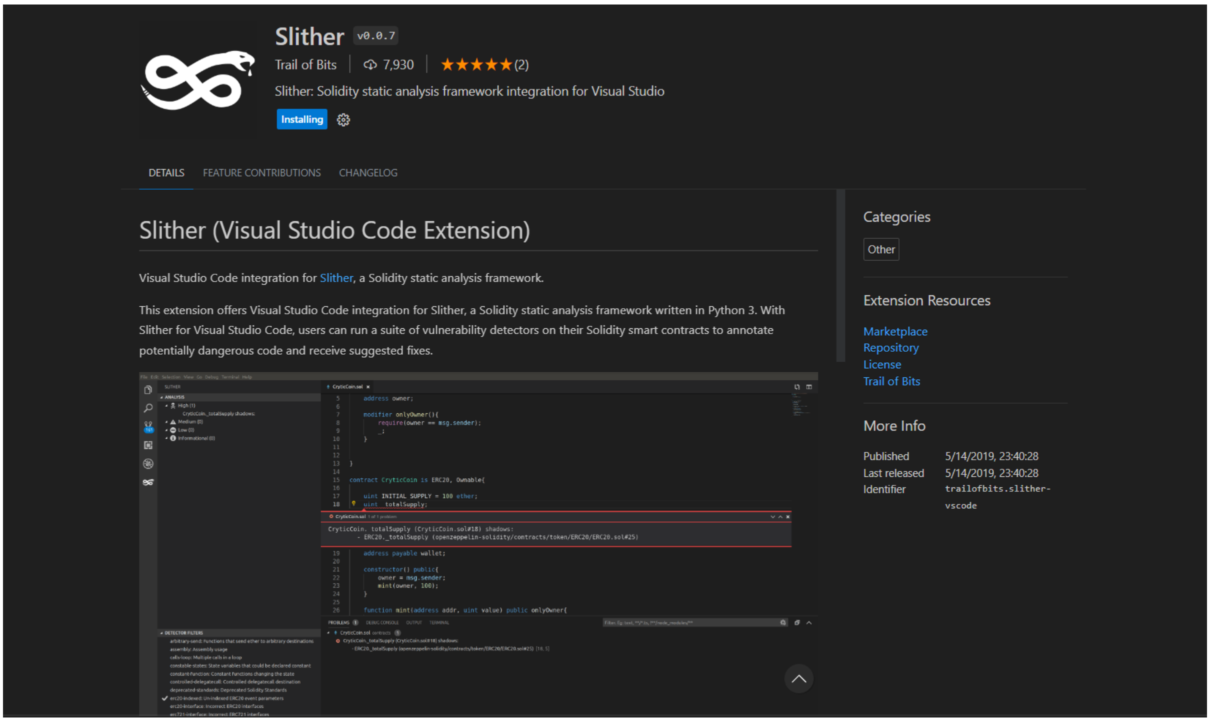


```
Error: Source "interfaces/IEigenPodManager.sol" not found: File not found. Searched the following locations: "".
 --> StrategyManagerStorage.sol:6:1:
  |
6 | import "../interfaces/IEigenPodManager.sol";
  | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Error: Source "interfaces/IDelegationManager.sol" not found: File not found. Searched the following locations: "".
 --> StrategyManagerStorage.sol:7:1:
  |
7 | import "../interfaces/IDelegationManager.sol";
  | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Error: Source "interfaces/ISlasher.sol" not found: File not found. Searched the following locations: "".
 --> StrategyManagerStorage.sol:8:1:
  |
8 | import "../interfaces/ISlasher.sol";
  | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

I tried every possible fix to solve the following error. But I couldn't come across the error that kept popping up. So transferred all my files to the Windows operating system and installed the plugging Slither to visual studio code.

5. Slither on visual studio code.

I installed the vs code plugging to the vs code environment.



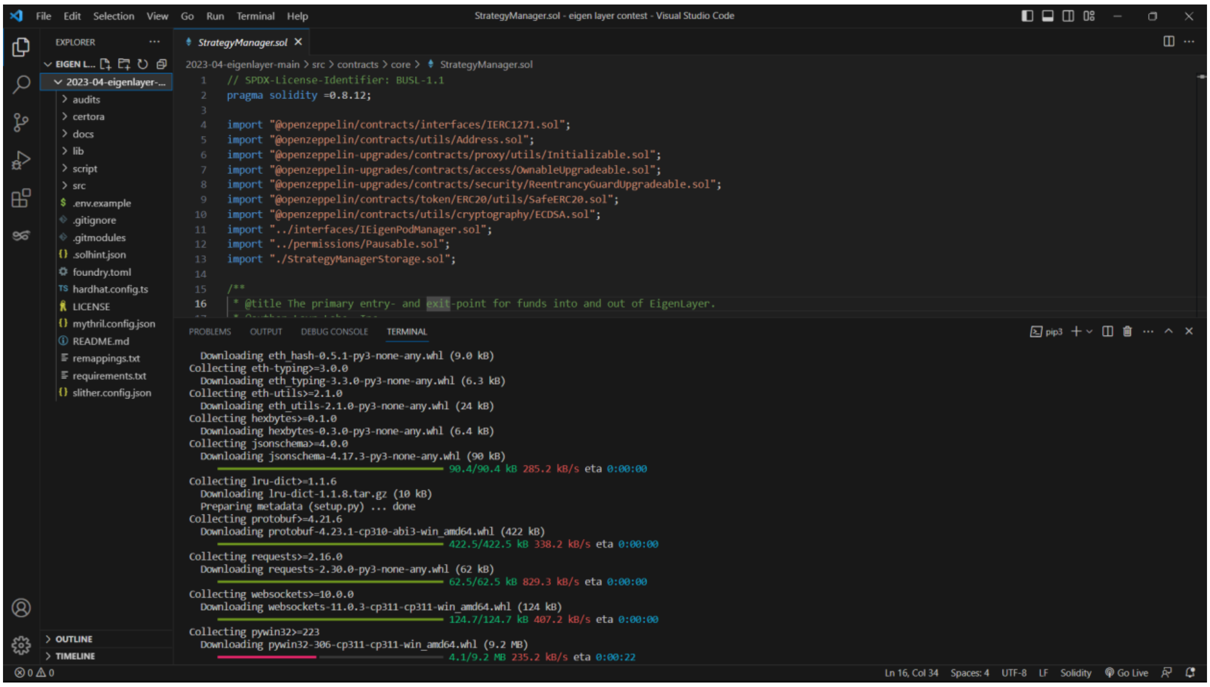
Slither: Solidity static analysis framework integration for Visual Studio

Slither (Visual Studio Code Extension)

Visual Studio Code integration for Slither, a Solidity static analysis framework.

This extension offers Visual Studio Code integration for Slither, a Solidity static analysis framework written in Python 3. With Slither for Visual Studio Code, users can run a suite of vulnerability detectors on their Solidity smart contracts to annotate potentially dangerous code and receive suggested fixes.

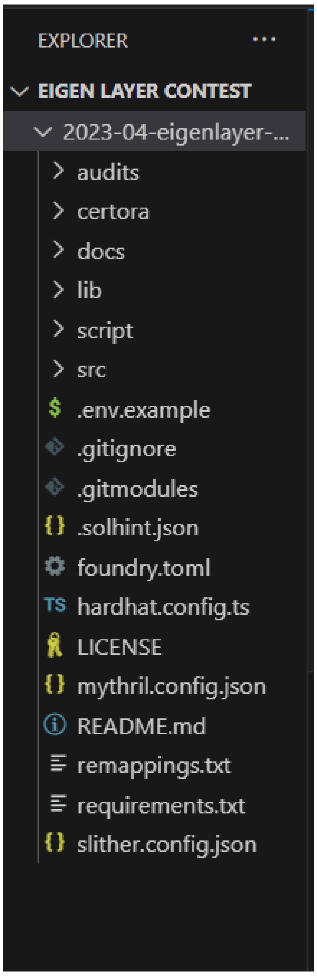
Published	5/14/2019, 23:40:28
Last released	5/14/2019, 23:40:28
Identifier	trailofbits.slither-vscode



```
// SPDX-License-Identifier: BUSL-1.1
pragma solidity =0.8.12;

import "@openzeppelin/contracts/interfaces/IERC1271.sol";
import "@openzeppelin/contracts/utils/Address.sol";
import "@openzeppelin-upgrades/contracts/proxy/utils/Initializable.sol";
import "@openzeppelin-upgrades/contracts/access/OwnableUpgradeable.sol";
import "@openzeppelin-upgrades/contracts/security/ReentrancyGuardUpgradeable.sol";
import "@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol";
import "@openzeppelin/contracts/utils/cryptography/ECDSA.sol";
import "../interfaces/IEigenPodManager.sol";
import "../permissions/Pausable.sol";
import "./StrategyManagerStorage.sol";

/**
 * @title The primary entry- and exit-point for funds into and out of EigenLayer.
```

```
  Downloading eth_hash-0.5.1-py3-none-any.whl (9.0 kB)
Collecting eth-typing>=3.0.0
  Downloading eth_typing-3.3.0-py3-none-any.whl (6.3 kB)
Collecting eth-utils>=2.1.0
  Downloading eth_utils-2.1.0-py3-none-any.whl (24 kB)
Collecting hexbytes>=0.1.0
  Downloading hexbytes-0.3.0-py3-none-any.whl (6.4 kB)
Collecting jsonschema>=4.0.0
  Downloading jsonschema-4.17.3-py3-none-any.whl (90 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 90.4/90.4 kB 285.2 kB/s eta 0:00:00
Collecting lru-dict>=1.1.6
  Downloading lru-dict-1.1.8.tar.gz (10 kB)
  Preparing metadata (setup.py) ... done
Collecting protobuf>=4.21.6
  Downloading protobuf-4.23.1-cp310-abi3-win_amd64.whl (422 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 422.5/422.5 kB 338.2 kB/s eta 0:00:00
Collecting requests>=2.16.0
  Downloading requests-2.30.0-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.5/62.5 kB 829.3 kB/s eta 0:00:00
Collecting websockets>=10.0.0
  Downloading websockets-11.0.3-cp311-cp311-win_amd64.whl (124 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.7/124.7 kB 407.2 kB/s eta 0:00:00
Collecting pywin32>=223
  Downloading pywin32-306-cp311-cp311-win_amd64.whl (9.2 MB)
     ━━━━━━━━━━━━━━━━──────────────────────── 4.1/9.2 MB 235.2 kB/s eta 0:00:22
```

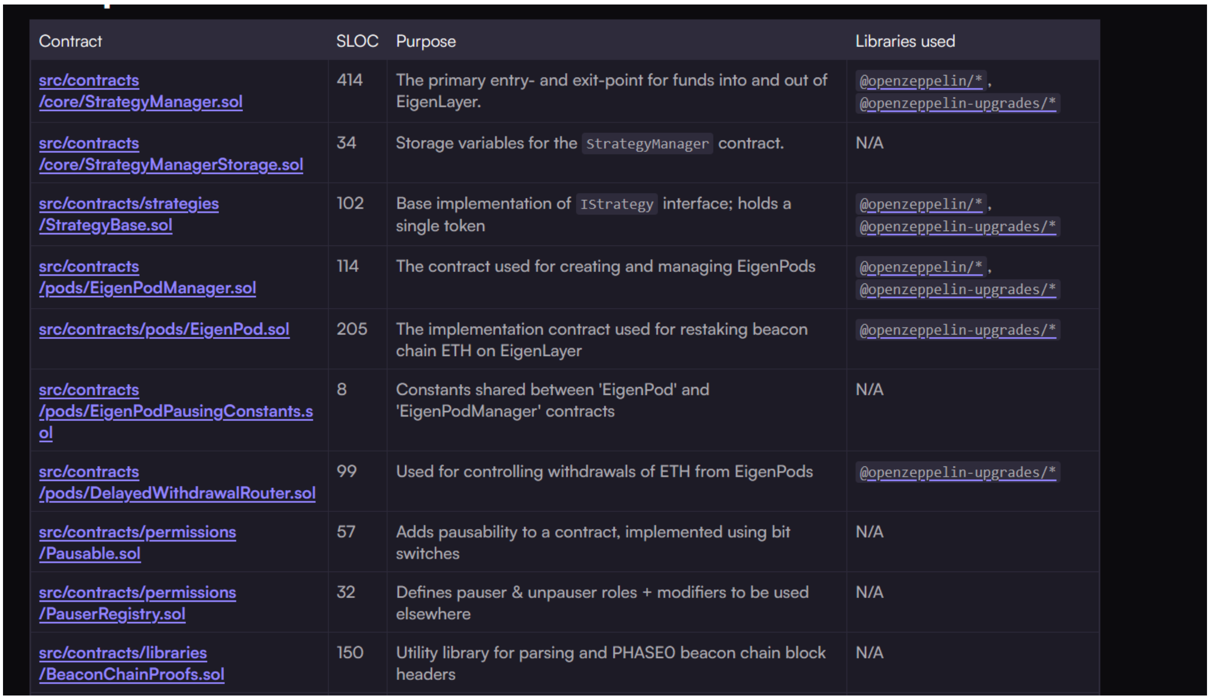
Then opened the Eiger layer competition files to and started analyzing vulnerabilities in the respective in scope sol files.



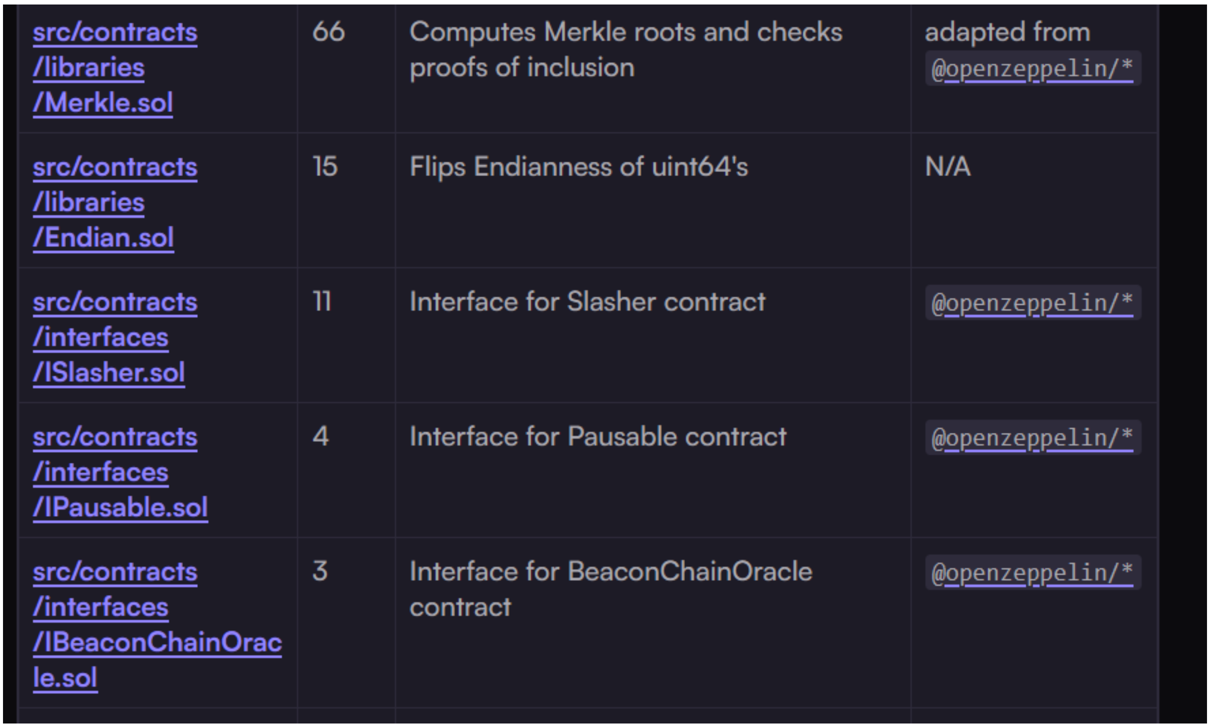
```
EXPLORER ···
∨ EIGEN LAYER CONTEST
  ∨ 2023-04-eigenlayer-...
    > audits
    > certora
    > docs
    > lib
    > script
    > src
    $ .env.example
      .gitignore
      .gitmodules
    {} .solhint.json
      foundry.toml
    TS hardhat.config.ts
      LICENSE
    {} mythril.config.json
      README.md
      remappings.txt
      requirements.txt
    {} slither.config.json
```

6. Analyzing sol files for vulnerabilities.

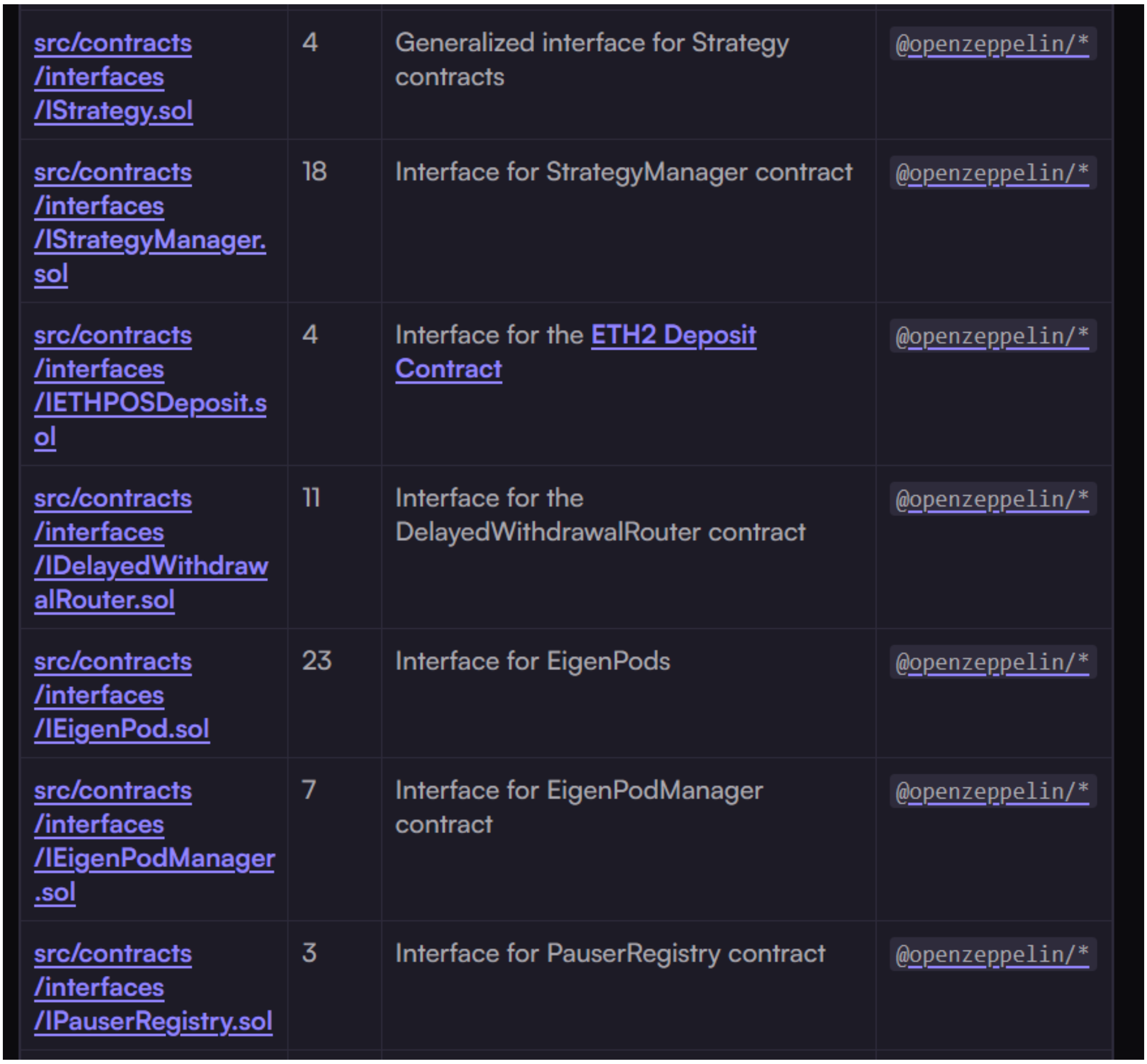
i. Sol file that are in scope



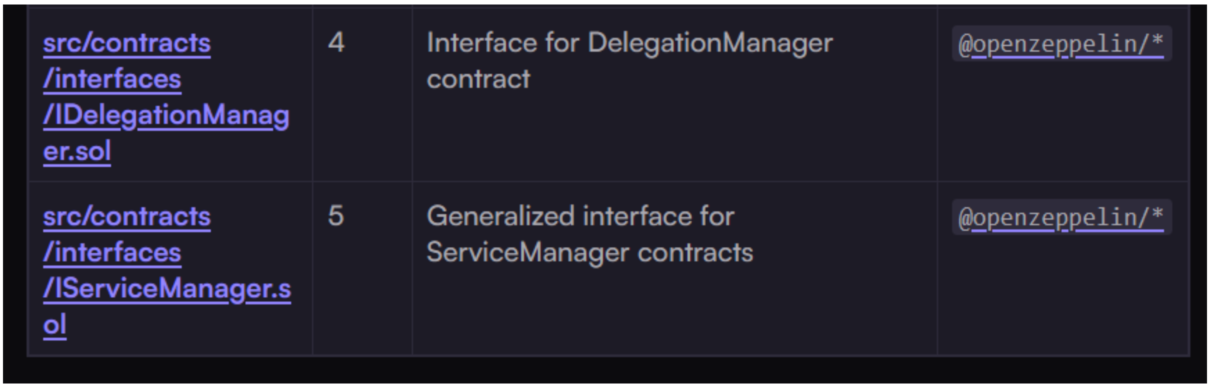
Contract	SLOC	Purpose	Libraries used
src/contracts /core/StrategyManager.sol	414	The primary entry- and exit-point for funds into and out of EigenLayer.	@openzeppelin/* , @openzeppelin-upgrades/*
src/contracts /core/StrategyManagerStorage.sol	34	Storage variables for the `StrategyManager` contract.	N/A
src/contracts/strategies /StrategyBase.sol	102	Base implementation of `IStrategy` interface; holds a single token	@openzeppelin/* , @openzeppelin-upgrades/*
src/contracts /pods/EigenPodManager.sol	114	The contract used for creating and managing EigenPods	@openzeppelin/* , @openzeppelin-upgrades/*
src/contracts/pods/EigenPod.sol	205	The implementation contract used for restaking beacon chain ETH on EigenLayer	@openzeppelin-upgrades/*
src/contracts /pods/EigenPodPausingConstants.sol	8	Constants shared between 'EigenPod' and 'EigenPodManager' contracts	N/A
src/contracts /pods/DelayedWithdrawalRouter.sol	99	Used for controlling withdrawals of ETH from EigenPods	@openzeppelin-upgrades/*
src/contracts/permissions /Pausable.sol	57	Adds pausability to a contract, implemented using bit switches	N/A
src/contracts/permissions /PauserRegistry.sol	32	Defines pauser & unpauser roles + modifiers to be used elsewhere	N/A
src/contracts/libraries /BeaconChainProofs.sol	150	Utility library for parsing and PHASE0 beacon chain block headers	N/A



src/contracts /libraries /Merkle.sol	66	Computes Merkle roots and checks proofs of inclusion	adapted from @openzeppelin/*
src/contracts /libraries /Endian.sol	15	Flips Endianness of uint64's	N/A
src/contracts /interfaces /ISlasher.sol	11	Interface for Slasher contract	@openzeppelin/*
src/contracts /interfaces /IPausable.sol	4	Interface for Pausable contract	@openzeppelin/*
src/contracts /interfaces /IBeaconChainOracle.sol	3	Interface for BeaconChainOracle contract	@openzeppelin/*



src/contracts/interfaces/IStrategy.sol	4	Generalized interface for Strategy contracts	`@openzeppelin/*`
src/contracts/interfaces/IStrategyManager.sol	18	Interface for StrategyManager contract	`@openzeppelin/*`
src/contracts/interfaces/IETHPOSDeposit.sol	4	Interface for the ETH2 Deposit Contract	`@openzeppelin/*`
src/contracts/interfaces/IDelayedWithdrawalRouter.sol	11	Interface for the DelayedWithdrawalRouter contract	`@openzeppelin/*`
src/contracts/interfaces/IEigenPod.sol	23	Interface for EigenPods	`@openzeppelin/*`
src/contracts/interfaces/IEigenPodManager.sol	7	Interface for EigenPodManager contract	`@openzeppelin/*`
src/contracts/interfaces/IPauserRegistry.sol	3	Interface for PauserRegistry contract	`@openzeppelin/*`

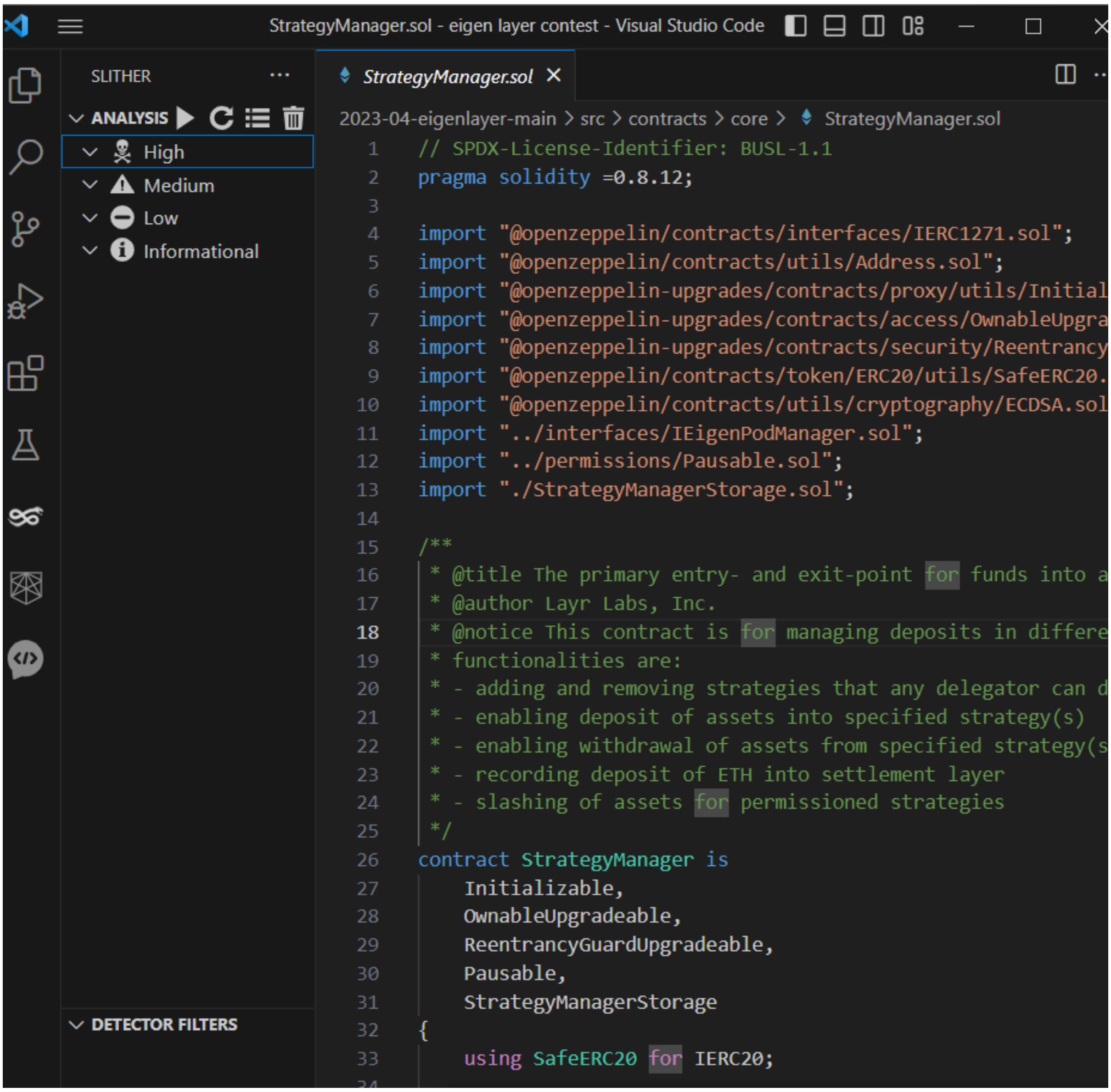


src/contracts/interfaces/IDelegationManager.sol	4	Interface for DelegationManager contract	`@openzeppelin/*`
src/contracts/interfaces/IServiceManager.sol	5	Generalized interface for ServiceManager contracts	`@openzeppelin/*`

ii. Vulnerability assessment

Then I analyzed each in-scope sol file using slither analyzer obtained vulnerabilities.

Example snapshot given below.



StrategyManager.sol - eigen layer contest - Visual Studio Code
SLITHER
ANALYSIS
High
Medium
Low
Informational
DETECTOR FILTERS
StrategyManager.sol
2023-04-eigenlayer-main > src > contracts > core > StrategyManager.sol

```
// SPDX-License-Identifier: BUSL-1.1
pragma solidity =0.8.12;

import "@openzeppelin/contracts/interfaces/IERC1271.sol";
import "@openzeppelin/contracts/utils/Address.sol";
import "@openzeppelin-upgrades/contracts/proxy/utils/Initial
import "@openzeppelin-upgrades/contracts/access/OwnableUpgra
import "@openzeppelin-upgrades/contracts/security/Reentrancy
import "@openzeppelin/contracts/token/ERC20/utils/SafeERC20.
import "@openzeppelin/contracts/utils/cryptography/ECDSA.sol
import "../interfaces/IEigenPodManager.sol";
import "../permissions/Pausable.sol";
import "./StrategyManagerStorage.sol";

/**
 * @title The primary entry- and exit-point for funds into a
 * @author Layr Labs, Inc.
 * @notice This contract is for managing deposits in differe
 * functionalities are:
 * - adding and removing strategies that any delegator can d
 * - enabling deposit of assets into specified strategy(s)
 * - enabling withdrawal of assets from specified strategy(s
 * - recording deposit of ETH into settlement layer
 * - slashing of assets for permissioned strategies
 */
contract StrategyManager is
    Initializable,
    OwnableUpgradeable,
    ReentrancyGuardUpgradeable,
    Pausable,
    StrategyManagerStorage
{
    using SafeERC20 for IERC20;
```

1. StratergyManagerStorage.sol

```
INFO:Detectors:
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/core/StrategyManagerStorage.sol#2)
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5]) (contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21]) (contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3]) (contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8]) (contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32) (contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32) (contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[]) (contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes32[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used and should be removed
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256) (contracts/libraries/Merkle.sol#29-36) is never used and should be removed
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256) (contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/core/StrategyManagerStorage.sol#2) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
```

```
Variable StrategyManagerStorage.DOMAIN_SEPARATOR (contracts/core/StrategyManagerStorage.sol#23) is not in mixedCase
Variable StrategyManagerStorage.__gap (contracts/core/StrategyManagerStorage.sol#83) is not in mixedCase
Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not in mixedCase
Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not in mixedCase
Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in CapWords
Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) | ((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 & n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56) (contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
StrategyManagerStorage (contracts/core/StrategyManagerStorage.sol#15-84) does not implement functions:
    - IStrategyManager.addStrategiesToDepositWhitelist(IStrategy[]) (contracts/interfaces/IStrategyManager.sol#210)
    - IStrategyManager.calculateWithdrawalRoot(IStrategyManager.QueuedWithdrawal) (contracts/interfaces/IStrategyManager.sol#202-207)
    - IStrategyManager.completeQueuedWithdrawal(IStrategyManager.QueuedWithdrawal,IERC20[],uint256,bool) (contracts/interfaces/IStrategyManager.sol#146-152)
    - IStrategyManager.completeQueuedWithdrawals(IStrategyManager.QueuedWithdrawal[],IERC20[][],uint256[],bool[]) (contracts/interfaces/IStrategyManager.sol#159-165)
    - IStrategyManager.depositBeaconChainETH(address,uint256) (contracts/interfaces/IStrategyManager.sol#55)
    - IStrategyManager.depositIntoStrategy(IStrategy,IERC20,uint256) (contracts/interfaces/IStrategyManager.sol#43-45)
    - IStrategyManager.depositIntoStrategyWithSignature(IStrategy,IERC20,uint256,address,uint256,bytes) (contracts/interfaces/IStrategyManager.sol#82-91)
    - IStrategyManager.getDeposits(address) (contracts/interfaces/IStrategyManager.sol#100)
    - IStrategyManager.queueWithdrawal(uint256[],IStrategy[],uint256[],address,bool) (contracts/interfaces/IStrategyManager.sol#126-133)
    - IStrategyManager.recordOvercommittedBeaconChainETH(address,uint256,uint256) (contracts/interfaces/IStrategyManager.sol#64-65)
    - IStrategyManager.removeStrategiesFromDepositWhitelist(IStrategy[]) (contracts/interfaces/IStrategyManager.sol#213)
```

```
	- IStrategyManager.slashQueuedWithdrawal(address,IStrategyManager.QueuedWithdrawal,IERC20[],uint256[]) (contracts/interfaces/IStrategyManager.sol#198-199)
	- IStrategyManager.slashShares(address,address,IStrategy[],IERC20[],uint256[],uint256[]) (contracts/interfaces/IStrategyManager.sol#178-186)
	- IStrategyManager.stakerStrategyListLength(address) (contracts/interfaces/IStrategyManager.sol#103)
	- IStrategyManager.stakerStrategyShares(address,IStrategy) (contracts/interfaces/IStrategyManager.sol#94)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
StrategyManagerStorage.MAX_STAKER_STRATEGY_LIST_LENGTH (contracts/core/StrategyManagerStorage.sol#28) is never used in StrategyManagerStorage (contracts/core/StrategyManagerStorage.sol#15-84)
StrategyManagerStorage.__gap (contracts/core/StrategyManagerStorage.sol#83) is never used in StrategyManagerStorage (contracts/core/StrategyManagerStorage.sol#15-84)
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS (contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS (contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS (contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX (contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX (contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX (contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX (contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Detectors:
StrategyManagerStorage.DOMAIN_SEPARATOR (contracts/core/StrategyManagerStorage.sol#23) should be constant
StrategyManagerStorage.strategyWhitelister (contracts/core/StrategyManagerStorage.sol#36) should be constant
StrategyManagerStorage.withdrawalDelayBlocks (contracts/core/StrategyManagerStorage.sol#44) should be constant
```

```
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
    INFO:Slither:contracts/core/StrategyManagerStorage.sol analyzed (15 contracts with 85 detectors), 71 result(s) found

```

2. StratergyManager.sol

```
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse = (3 * denominator) ^ 2 (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#117)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#121)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#122)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#123)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#124)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#125)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - denominator = denominator / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
    - inverse *= 2 - denominator * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#126)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
    - prod0 = prod0 / twos (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#105)
    - result = prod0 * inverse (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Reentrancy in StrategyManager._completeQueuedWithdrawal(IStrategyManager.QueuedWithdrawal,IERC20[],uint256,bool) (contracts/core/StrategyManager.sol#745-805):
    External calls:
    - require(bool,string)(slasher.canWithdraw(queuedWithdrawal.delegatedAddress,queuedWithdrawal.withdrawalStartBlock,middlewareTimesIndex),StrategyManager.completeQueuedWithdrawal: shares pending withdrawal are still slashable) (contracts/core/StrategyManager.sol#755-758)
    State variables written after the call(s):
    - withdrawalRootPending[withdrawalRoot] = false (contracts/core/StrategyManager.sol#772)
    StrategyManagerStorage.withdrawalRootPending (contracts/core/StrategyManagerStorage.sol#53) can be used in cross function reentrancies:
    - StrategyManagerStorage.withdrawalRootPending (contracts/core/StrategyManagerStorage.sol#53)
Reentrancy in StrategyManager._completeQueuedWithdrawal(IStrategyManager.QueuedWithdrawal,IERC20[],uint256,bool) (contracts/core/StrategyManager.sol#745-805):
    External calls:
    - require(bool,string)(slasher.canWithdraw(queuedWithdrawal.delegatedAddress,queuedWithdrawal.withdrawalStartBlock,middlewareTimesIndex),StrategyManager.completeQueuedWithdrawal: shares pending withdrawal are still slashable) (contracts/core/StrategyManager.sol#755-758)
    - _withdrawBeaconChainETH(queuedWithdrawal.depositor,msg.sender,queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#784)
        - eigenPodManager.withdrawRestakedBeaconChainETH(staker,recipient,amount) (contracts/core/StrategyManager.sol#835)
    - queuedWithdrawal.strategies[i].withdraw(msg.sender,tokens[i],queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#787-789)
    State variables written after the call(s):
    - _withdrawBeaconChainETH(queuedWithdrawal.depositor,msg.sender,queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#784)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = 0 (contracts/core/StrategyManager.sol#825)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = (amountToDecrement - amount) (contracts/core/StrategyManager.sol#829)
    StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68) can be used in cross function reentrancies:
    - StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68)
Reentrancy in StrategyManager.slashQueuedWithdrawal(address,IStrategyManager.QueuedWithdrawal,IERC20[],uint256[]) (contracts/core/StrategyManager.sol#536-579):
    External calls:
    - _withdrawBeaconChainETH(queuedWithdrawal.depositor,recipient,queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#569)
        - eigenPodManager.withdrawRestakedBeaconChainETH(staker,recipient,amount) (contracts/core/StrategyManager.sol#835)
    - queuedWithdrawal.strategies[i].withdraw(recipient,tokens[i],queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#572)
    State variables written after the call(s):
    - _withdrawBeaconChainETH(queuedWithdrawal.depositor,recipient,queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#569)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = 0 (contracts/core/StrategyManager.sol#825)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = (amountToDecrement - amount) (contracts/core/StrategyManager.sol#829)
    StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68) can be used in cross function reentrancies:
    - StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68)
Reentrancy in StrategyManager.slashShares(address,address,IStrategy[],IERC20[],uint256[],uint256[]) (contracts/core/StrategyManager.sol#482-524):
    External calls:
    - _withdrawBeaconChainETH(slashedAddress,recipient,shareAmounts[i]) (contracts/core/StrategyManager.sol#509)
        - eigenPodManager.withdrawRestakedBeaconChainETH(staker,recipient,amount) (contracts/core/StrategyManager.sol#835)
    - strategies[i].withdraw(recipient,tokens[i],shareAmounts[i]) (contracts/core/StrategyManager.sol#513)
    State variables written after the call(s):
```

```
    - _withdrawBeaconChainETH(slashedAddress,recipient,shareAmounts[i]) (contracts/core/StrategyManager.sol#509)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = 0 (contracts/core/StrategyManager.sol#825)
        - beaconChainETHSharesToDecrementOnWithdrawal[staker] = (amountToDecrement - amount) (contracts/core/StrategyManager.sol#829)
    StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68) can be used in cross function reentrancies:
    - StrategyManagerStorage.beaconChainETHSharesToDecrementOnWithdrawal (contracts/core/StrategyManagerStorage.sol#68)
    - _removeShares(slashedAddress,strategyIndexes[strategyIndexIndex],strategies[i],shareAmounts[i]) (contracts/core/StrategyManager.sol#501)
        - stakerStrategyList[depositor][strategyIndex] = stakerStrategyList[depositor][stakerStrategyList[depositor].length - 1] (contracts/core/StrategyManager.sol#719-720)
        - stakerStrategyList[depositor][j] = stakerStrategyList[depositor][stakerStrategyList[depositor].length - 1] (contracts/core/StrategyManager.sol#728)
        - stakerStrategyList[depositor].pop() (contracts/core/StrategyManager.sol#739)
    StrategyManagerStorage.stakerStrategyList (contracts/core/StrategyManagerStorage.sol#51) can be used in cross function reentrancies:
    - StrategyManager._undelegate(address) (contracts/core/StrategyManager.sol#811-814)
    - StrategyManager.getDeposits(address) (contracts/core/StrategyManager.sol#857-868)
    - StrategyManagerStorage.stakerStrategyList (contracts/core/StrategyManagerStorage.sol#51)
    - StrategyManager.stakerStrategyListLength(address) (contracts/core/StrategyManager.sol#871-873)
    - _removeShares(slashedAddress,strategyIndexes[strategyIndexIndex],strategies[i],shareAmounts[i]) (contracts/core/StrategyManager.sol#501)
        - stakerStrategyShares[depositor][strategy] = userShares (contracts/core/StrategyManager.sol#697)
    StrategyManagerStorage.stakerStrategyShares (contracts/core/StrategyManagerStorage.sol#49) can be used in cross function reentrancies:
    - StrategyManager.getDeposits(address) (contracts/core/StrategyManager.sol#857-868)
    - StrategyManagerStorage.stakerStrategyShares (contracts/core/StrategyManagerStorage.sol#49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
StrategyManager.slashShares(address,address,IStrategy[],IERC20[],uint256[],uint256[]).strategyIndexIndex (contracts/core/StrategyManager.sol#496) is a local variable never initialized
StrategyManager.queueWithdrawal(uint256[],IStrategy[],uint256[],address,bool).strategyIndexIndex (contracts/core/StrategyManager.sol#351) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
StrategyManager.onlyNotFrozen(address) (contracts/core/StrategyManager.sol#96-102) has external calls inside a loop: require(bool,string)(!slasher.isFrozen(staker),StrategyManager.onlyNotFrozen: staker has been frozen and may be subject to slashing) (contracts/core/StrategyManager.sol#97-100)
StrategyManager._completeQueuedWithdrawal(IStrategyManager.QueuedWithdrawal,IERC20[],uint256,bool) (contracts/core/StrategyManager.sol#745-805) has external calls inside a loop: require(bool,string)(slasher.canWithdraw(queuedWithdrawal.delegatedAddress,queuedWithdrawal.withdrawalStartBlock,middlewareTimesIndex),StrategyManager.completeQueuedWithdrawal: shares pending withdrawal are still slashable) (contracts/core/StrategyManager.sol#755-758)
StrategyManager._withdrawBeaconChainETH(address,address,uint256) (contracts/core/StrategyManager.sol#821-836) has external calls inside a loop: eigenPodManager.withdrawRestakedBeaconChainETH(staker,recipient,amount) (contracts/core/StrategyManager.sol#835)
StrategyManager._completeQueuedWithdrawal(IStrategyManager.QueuedWithdrawal,IERC20[],uint256,bool) (contracts/core/StrategyManager.sol#745-805) has external calls inside a loop: queuedWithdrawal.strategies[i].withdraw(msg.sender,tokens[i],queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#787-789)
StrategyManager._addShares(address,IStrategy,uint256) (contracts/core/StrategyManager.sol#629-648) has external calls inside a loop: delegation.increaseDelegatedShares(depositor,strategy,shares) (contracts/core/StrategyManager.sol#647)
StrategyManager.slashShares(address,address,IStrategy[],IERC20[],uint256[],uint256[]) (contracts/core/StrategyManager.sol#482-524) has external calls inside a loop: strategies[i].withdraw(recipient,tokens[i],shareAmounts[i]) (contracts/core/StrategyManager.sol#513)
StrategyManager.slashQueuedWithdrawal(address,IStrategyManager.QueuedWithdrawal,IERC20[],uint256[]) (contracts/core/StrategyManager.sol#536-579) has external calls inside a loop: queuedWithdrawal.strategies[i].withdraw(recipient,tokens[i],queuedWithdrawal.shares[i]) (contracts/core/StrategyManager.sol#572)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop
INFO:Detectors:
Reentrancy in StrategyManager.queueWithdrawal(uint256[],IStrategy[],uint256[],address,bool) (contracts/core/StrategyManager.sol#329-429):
	External calls:
	- delegation.decreaseDelegatedShares(msg.sender,strategies,shares) (contracts/core/StrategyManager.sol#346)
	State variables written after the call(s):
	- numWithdrawalsQueued[msg.sender] = nonce + 1 (contracts/core/StrategyManager.sol#396)
	- _removeShares(msg.sender,strategyIndexes[strategyIndexIndex],strategies[i],shares[i]) (contracts/core/StrategyManager.sol#370)
		- stakerStrategyList[depositor][strategyIndex] = stakerStrategyList[depositor][stakerStrategyList[depositor].length - 1] (contracts/core/StrategyManager.sol#719-720)
		- stakerStrategyList[depositor][j] = stakerStrategyList[depositor][stakerStrategyList[depositor].length - 1] (contracts/core/StrategyManager.sol#728)
		- stakerStrategyList[depositor].pop() (contracts/core/StrategyManager.sol#739)
	- _removeShares(msg.sender,strategyIndexes[strategyIndexIndex],strategies[i],shares[i]) (contracts/core/StrategyManager.sol#370)
		- stakerStrategyShares[depositor][strategy] = userShares (contracts/core/StrategyManager.sol#697)
	- withdrawalRootPending[withdrawalRoot] = true (contracts/core/StrategyManager.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
StrategyManager.depositIntoStrategyWithSignature(IStrategy,IERC20,uint256,address,uint256,bytes) (contracts/core/StrategyManager.sol#248-298) uses timestamp for comparisons
	Dangerous comparisons:
	- require(bool,string)(expiry >= block.timestamp,StrategyManager.depositIntoStrategyWithSignature: signature expired) (contracts/core/StrategyManager.sol#262-265)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address._revert(bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) uses assembly
	- INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#236-239)
Strings.toString(uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#18-38) uses assembly
	- INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#24-26)
	- INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#30-32)
ECDSA.tryRecover(bytes32,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#55-72) uses assembly
	- INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#63-67)
Math.mulDiv(uint256,uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) uses assembly
```

```
    - INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#66-70)
    - INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#86-93)
    - INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#100-109)
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256) (contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0', '^0.8.1', '^0.8.2']
    - =0.8.12 (contracts/core/StrategyManager.sol#2)
    - =0.8.12 (contracts/core/StrategyManagerStorage.sol#2)
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - =0.8.12 (contracts/permissions/Pausable.sol#3)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/interfaces/IERC1271.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4)
```

```
        - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#4)
        - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#4)
        - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.2 (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
    INFO:Detectors:
    Address.functionCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
    Address.functionCallWithValue(address,bytes,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
    Address.functionDelegateCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#170-172) is never used and should be removed
    Address.functionDelegateCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187) is never used and should be removed
    Address.functionStaticCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#145-147) is never used and should be removed
    Address.functionStaticCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162) is never used and should be removed
    Address.sendValue(address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
    Address.verifyCallResult(bool,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#219-229) is never used and should be removed
    BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5]) (contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
    BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21]) (contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
    BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3]) (contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
    BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8]) (contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
```

```
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32) (contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32) (contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[]) (contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes32[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be removed
Context._msgData() (contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ECDSA.recover(bytes32,bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#116-124) is never used and should be removed
ECDSA.recover(bytes32,uint8,bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#164-173) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#197-199) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes32) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#183-187) is never used and should be removed
ECDSA.toTypedDataHash(bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#210-212) is never used and should be removed
ECDSA.tryRecover(bytes32,bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#101-109) is never used and should be removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used and should be removed
Initializable._getInitializedVersion() (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#162-164) is never used and should be removed
Math.average(uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#34-37) is never used and should be removed
Math.ceilDiv(uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#45-48) is never used and should be removed
Math.log10(uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#258-290) is never used and should be removed
```

```
199.      Math.log10(uint256,Math.Rounding)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#296-301) is
   never used and should be removed
200.      Math.log2(uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#205-241) is
   never used and should be removed
201.      Math.log2(uint256,Math.Rounding)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#247-252) is
   never used and should be removed
202.      Math.log256(uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#309-333) is
   never used and should be removed
203.      Math.log256(uint256,Math.Rounding)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#339-344) is
   never used and should be removed
204.      Math.max(uint256,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#19-21) is
   never used and should be removed
205.      Math.min(uint256,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#26-28) is
   never used and should be removed
206.      Math.mulDiv(uint256,uint256,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) is
   never used and should be removed
207.      Math.mulDiv(uint256,uint256,uint256,Math.Rounding)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#140-151) is
   never used and should be removed
208.      Math.sqrt(uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#158-189) is
   never used and should be removed
209.      Math.sqrt(uint256,Math.Rounding)
   (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#194-199) is
   never used and should be removed
210.      Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is
   never used and should be removed
211.      Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
   (contracts/libraries/Merkle.sol#48-70) is never used and should be removed
212.      Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
   (contracts/libraries/Merkle.sol#99-121) is never used and should be removed
213.      Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
   (contracts/libraries/Merkle.sol#29-36) is never used and should be removed
214.      Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
   (contracts/libraries/Merkle.sol#80-87) is never used and should be removed
215.      SafeERC20.safeApprove(IERC20,address,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
   #46-59) is never used and should be removed
216.      SafeERC20.safeDecreaseAllowance(IERC20,address,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
   #70-81) is never used and should be removed
217.      SafeERC20.safeIncreaseAllowance(IERC20,address,uint256)
   (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
   #61-68) is never used and should be removed
```

```
SafeERC20.safePermit(IERC20Permit,address,address,uint256,uint256,uint8,bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#83-97) is never used and should be removed
SafeERC20.safeTransfer(IERC20,address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#22-28) is never used and should be removed
Strings.toHexString(address) (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#67-69) is never used and should be removed
Strings.toHexString(uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#43-47) is never used and should be removed
Strings.toHexString(uint256,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#52-62) is never used and should be removed
Strings.toString(uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#18-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/core/StrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/core/StrategyManagerStorage.sol#2) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/interfaces/IERC1271.sol#4) allows old versions
Pragma version^0.8.2 (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
Pragma version=0.8.12 (contracts/permissions/Pausable.sol#3) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
    - (success) = recipient.call{value: amount}() (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137):
    - (success,returndata) = target.call{value: value}(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#135)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162):
        - (success,returndata) = target.staticcall(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#160)
    Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187):
        - (success,returndata) = target.delegatecall(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#185)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
    INFO:Detectors:
    Parameter StrategyManager.initialize(address,address,IPauserRegistry,uint256,uint256)._pauserRegistry (contracts/core/StrategyManager.sol#146) is not in mixedCase
    Parameter StrategyManager.initialize(address,address,IPauserRegistry,uint256,uint256)._withdrawalDelayBlocks (contracts/core/StrategyManager.sol#146) is not in mixedCase
    Parameter StrategyManager.setWithdrawalDelayBlocks(uint256)._withdrawalDelayBlocks (contracts/core/StrategyManager.sol#582) is not in mixedCase
    Variable StrategyManager.ORIGINAL_CHAIN_ID (contracts/core/StrategyManager.sol#42) is not in mixedCase
    Variable StrategyManagerStorage.DOMAIN_SEPARATOR (contracts/core/StrategyManagerStorage.sol#23) is not in mixedCase
    Variable StrategyManagerStorage.__gap (contracts/core/StrategyManagerStorage.sol#83) is not in mixedCase
    Function IERC20Permit.DOMAIN_SEPARATOR() (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#59) is not in mixedCase
    Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not in mixedCase
    Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not in mixedCase
    Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in CapWords
    Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44) is not in CapWords
    Variable Pausable.__gap (contracts/permissions/Pausable.sol#115) is not in mixedCase
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
    INFO:Detectors:
    Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals with too many digits:
        - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) | ((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 & n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56) (contracts/libraries/Endian.sol#10-18)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
    INFO:Detectors:
    StrategyManager (contracts/core/StrategyManager.sol#26-890) does not implement functions:
        - IStrategyManager.stakerStrategyShares(address,IStrategy) (contracts/interfaces/IStrategyManager.sol#94)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
    INFO:Detectors:
    Pausable.UNPAUSE_ALL (contracts/permissions/Pausable.sol#22) is never used in StrategyManager (contracts/core/StrategyManager.sol#26-890)
    Pausable.PAUSE_ALL (contracts/permissions/Pausable.sol#23) is never used in StrategyManager (contracts/core/StrategyManager.sol#26-890)
    BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS (contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS (contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS (contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
    BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX (contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
      BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX (contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX (contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX (contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
      Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
      INFO:Slither:contracts/core/StrategyManager.sol analyzed (28 contracts with 85 detectors), 158 result(s) found
```

3. StratergyBase.sol

```
INFO:Detectors:
StrategyBase.deposit(IERC20,uint256) (contracts/strategies/StrategyBase.sol#78-112) uses a dangerous strict equality:
    - priorTokenBalance == 0 (contracts/strategies/StrategyBase.sol#96)
StrategyBase.underlyingToSharesView(uint256) (contracts/strategies/StrategyBase.sol#196-203) uses a dangerous strict equality:
    - tokenBalance == 0 || totalShares == 0 (contracts/strategies/StrategyBase.sol#198)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
```

```
StrategyBase.deposit(IERC20,uint256) (contracts/strategies/StrategyBase.sol#78-112) should emit an event for:
    - totalShares = updatedTotalShares (contracts/strategies/StrategyBase.sol#110)
StrategyBase.withdraw(address,IERC20,uint256) (contracts/strategies/StrategyBase.sol#121-156) should emit an event for:
    - totalShares = updatedTotalShares (contracts/strategies/StrategyBase.sol#142)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Address._revert(bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) uses assembly
    - INLINE ASM (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#236-239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0', '^0.8.1', '^0.8.2']
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/permissions/Pausable.sol#3)
    - =0.8.12 (contracts/strategies/StrategyBase.sol#2)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4)
    - ^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4)
    - ^0.8.2 (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Address.functionCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#170-172) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#145-147) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162) is never used and should be removed
Address.sendValue(address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#219-229) is never used and should be removed
Initializable._getInitializedVersion() (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#162-164) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-59) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#70-81) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#61-68) is never used and should be removed
SafeERC20.safePermit(IERC20Permit,address,address,uint256,uint256,uint8,bytes32,bytes32) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#83-97) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#30-37) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.2 (contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4) allows old versions
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/permissions/Pausable.sol#3) allows old versions
Pragma version=0.8.12 (contracts/strategies/StrategyBase.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
    - (success) = recipient.call{value: amount}() (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137):
    - (success,returndata) = target.call{value: value}(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#135)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162):
    - (success,returndata) = target.staticcall(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#160)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187):
    - (success,returndata) = target.delegatecall(data) (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IERC20Permit.DOMAIN_SEPARATOR() (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#59) is not in mixedCase
Variable Pausable.__gap (contracts/permissions/Pausable.sol#115) is not in mixedCase
Parameter StrategyBase.initialize(IERC20,IPauserRegistry)._underlyingToken (contracts/strategies/StrategyBase.sol#51) is not in mixedCase
Parameter StrategyBase.initialize(IERC20,IPauserRegistry)._pauserRegistry (contracts/strategies/StrategyBase.sol#51) is not in mixedCase
Variable StrategyBase.__gap (contracts/strategies/StrategyBase.sol#250) is not in mixedCase
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Pausable.PAUSE_ALL (contracts/permissions/Pausable.sol#23) is never used in StrategyBase (contracts/strategies/StrategyBase.sol#19-251)
StrategyBase.__gap (contracts/strategies/StrategyBase.sol#250) is never used in StrategyBase (contracts/strategies/StrategyBase.sol#19-251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Slither:contracts/strategies/StrategyBase.sol analyzed (14 contracts with 85 detectors), 47 result(s) found

```

4. Permissions.sol

```
INFO:Detectors:
Pausable._initializePauser(IPauserRegistry,uint256)
(contracts/permissions/Pausable.sol#55-63) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/permissions/Pausable.sol#3) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Variable Pausable.__gap (contracts/permissions/Pausable.sol#115) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Pausable.UNPAUSE_ALL (contracts/permissions/Pausable.sol#22) is never used in Pausable
(contracts/permissions/Pausable.sol#15-116)
Pausable.PAUSE_ALL (contracts/permissions/Pausable.sol#23) is never used in Pausable
(contracts/permissions/Pausable.sol#15-116)
Pausable.__gap (contracts/permissions/Pausable.sol#115) is never used in Pausable
(contracts/permissions/Pausable.sol#15-116)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/permissions/PauserRegistry.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:contracts/permissions/ analyzed (5 contracts with 85 detectors), 12 result(s) found
```

5. Merkle.sol

```
INFO:Detectors:
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used
and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/libraries/Merkle.sol analyzed (1 contracts with 85 detectors), 9
result(s) found
```

6. IStrategyManager.sol

```
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
```

```
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IStrategyManager.sol analyzed (6 contracts with 85
detectors), 8 result(s) found
```

7. IStrategy.sol

```
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IStrategy.sol analyzed (2 contracts with 85 detectors),
4 result(s) found
```

8. ISlasher.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/ISlasher.sol analyzed (1 contracts with 85 detectors), 2
result(s) found
```

9. IServiceManager.sol

```
INFO:Detectors:
Different versions of Solidity are used:
	- Version used: ['=0.8.12', '^0.8.0']
	- =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
	- =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
	- =0.8.12 (contracts/interfaces/IServiceManager.sol#2)
	- =0.8.12 (contracts/interfaces/IStrategy.sol#2)
	- ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IServiceManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IServiceManager.sol analyzed (5 contracts with 85
detectors), 7 result(s) found
```

10.IPauseRegistry.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IPauserRegistry.sol analyzed (1 contracts with 85
detectors), 2 result(s) found
```

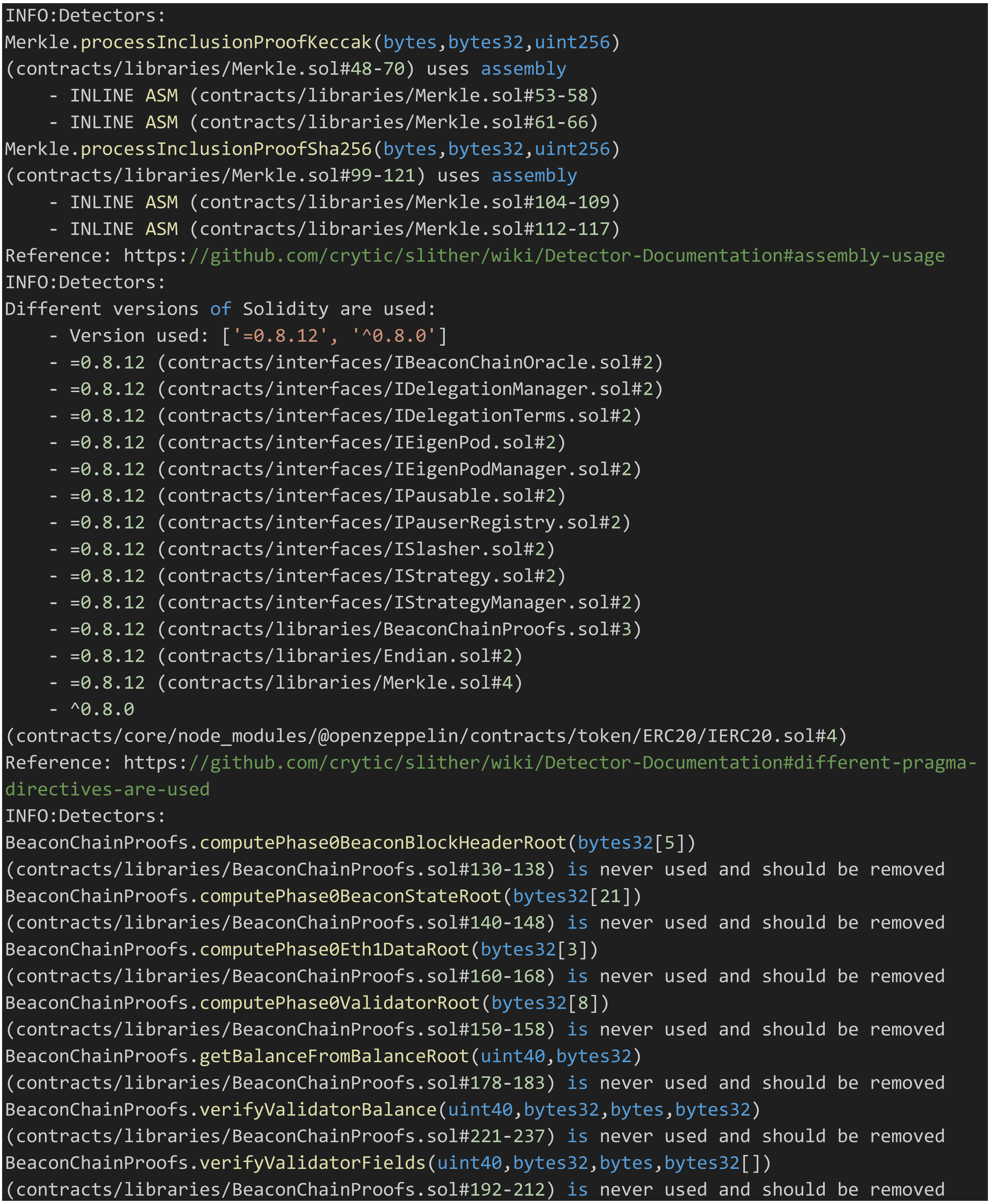
11.IPausable.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IPausable.sol analyzed (2 contracts with 85 detectors),
3 result(s) found
```

12.IETHPOSDeposite.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IETHPOSDeposit.sol#12) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Parameter IETHPOSDeposit.deposit(bytes,bytes,bytes,bytes32).withdrawal_credentials
(contracts/interfaces/IETHPOSDeposit.sol#29) is not in mixedCase
Parameter IETHPOSDeposit.deposit(bytes,bytes,bytes,bytes32).deposit_data_root
(contracts/interfaces/IETHPOSDeposit.sol#31) is not in mixedCase
Function IETHPOSDeposit.get_deposit_root() (contracts/interfaces/IETHPOSDeposit.sol#36) is
not in mixedCase
Function IETHPOSDeposit.get_deposit_count() (contracts/interfaces/IETHPOSDeposit.sol#40)
is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Slither:contracts/interfaces/IETHPOSDeposit.sol analyzed (1 contracts with 85
detectors), 6 result(s) found
```

13.IEigenPodManager.sol



```
INFO:Detectors:
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5])
(contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21])
(contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3])
(contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8])
(contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32)
(contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32)
(contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[])
(contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
```

```
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes3
2[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be
removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used
and should be removed
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used
and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not
in mixedCase
Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not
in mixedCase
Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in
CapWords
Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44)
is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS
(contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS
(contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS
(contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX
(contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX
(contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX
(contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX
(contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX
(contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX
(contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX
(contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable
INFO:Slither:contracts/interfaces/IEigenPodManager.sol analyzed (14 contracts with 85
detectors), 62 result(s) found
```

14.IEigenPod.sol

```
INFO:Detectors:
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5])
(contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21])
(contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3])
(contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8])
(contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32)
(contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32)
(contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[])
(contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes3
2[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be
removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used
and should be removed
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used
and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not
in mixedCase
Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not
in mixedCase
Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in
CapWords
Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44)
is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS
(contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS
(contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS
(contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX
(contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX
(contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX
(contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX
(contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX
(contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX
(contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX
(contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable
INFO:Slither:contracts/interfaces/IEigenPod.sol analyzed (14 contracts with 85 detectors),
62 result(s) found
```

15. IDelegationManager.sol

```
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:contracts/interfaces/IDelegationManager.sol analyzed (4 contracts with 85 detectors), 6 result(s) found
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0']
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:contracts/interfaces/IDelegationManager.sol analyzed (4 contracts with 85 detectors), 6 result(s) found
```

16.IdelayedWithdrawalRouter.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2) allows old
versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IDelayedWithdrawalRouter.sol analyzed (1 contracts with
85 detectors), 2 result(s) found
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2) allows old
versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IDelayedWithdrawalRouter.sol analyzed (1 contracts with
85 detectors), 2 result(s) found
```

17.IBeaconChainOracle.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IBeaconChainOracle.sol analyzed (1 contracts with 85
detectors), 2 result(s) found
INFO:Detectors:
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/interfaces/IBeaconChainOracle.sol analyzed (1 contracts with 85
detectors), 2 result(s) found
```

18.Endian.sol

```
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used
and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Slither:contracts/libraries/Endian.sol analyzed (1 contracts with 85 detectors), 4
result(s) found
```

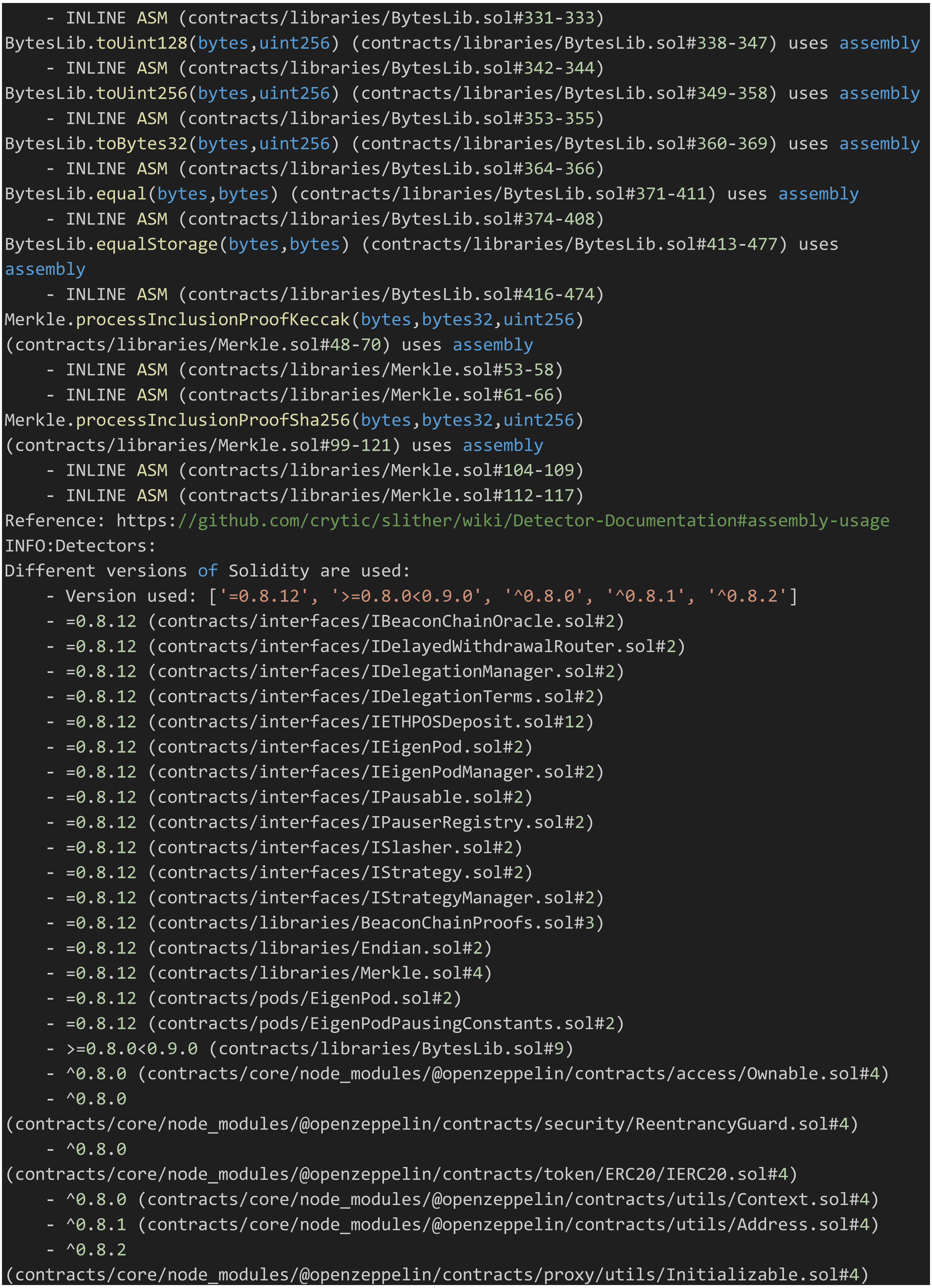
19.EigenPodPausingConstants.sol

```
INFO:Detectors:
Pragma version=0.8.12 (contracts/pods/EigenPodPausingConstants.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/pods/EigenPodPausingConstants.sol analyzed (1 contracts with 85
detectors), 2 result(s) found
INFO:Detectors:
Pragma version=0.8.12 (contracts/pods/EigenPodPausingConstants.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Slither:contracts/pods/EigenPodPausingConstants.sol analyzed (1 contracts with 85
detectors), 2 result(s) found
```

20.EigenPod.sol

```
INFO:Detectors:
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#82-211) performs a
multiplication on the result of a division:
    - sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 +
mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 **
32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2)
(contracts/libraries/BytesLib.sol#106-131)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#82-211) performs a
multiplication on the result of a division:
    -
sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) /
mask_concatStorage_asm_0 * mask_concatStorage_asm_0)
(contracts/libraries/BytesLib.sol#175)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#82-211) performs a
multiplication on the result of a division:
    -
sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) /
mask_concatStorage_asm_0 * mask_concatStorage_asm_0)
(contracts/libraries/BytesLib.sol#208)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#413-477) performs a
multiplication on the result of a division:
    - fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100
(contracts/libraries/BytesLib.sol#433)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-
multiply
INFO:Detectors:
EigenPod._processFullWithdrawal(uint64,uint40,uint256,address,IEigenPod.VALIDATOR_STATUS).
amountToSend (contracts/pods/EigenPod.sol#368) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-
local-variables
INFO:Detectors:
Reentrancy in
EigenPod._processFullWithdrawal(uint64,uint40,uint256,address,IEigenPod.VALIDATOR_STATUS)
(contracts/pods/EigenPod.sol#361-420):
    External calls:
    -
eigenPodManager.recordOvercommittedBeaconChainETH(podOwner,beaconChainETHStrategyIndex,uin
t256(REQUIRED_BALANCE_GWEI - withdrawalAmountGwei) * GWEI_TO_WEI)
(contracts/pods/EigenPod.sol#382)
    - eigenPodManager.restakeBeaconChainETH(podOwner,REQUIRED_BALANCE_WEI)
(contracts/pods/EigenPod.sol#396)
    - eigenPodManager.restakeBeaconChainETH(podOwner,uint256(withdrawalAmountGwei) *
GWEI_TO_WEI) (contracts/pods/EigenPod.sol#404)
    State variables written after the call(s):
    - validatorStatus[validatorIndex] = VALIDATOR_STATUS.WITHDRAWN
(contracts/pods/EigenPod.sol#412)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2
INFO:Detectors:
Reentrancy in
EigenPod._processFullWithdrawal(uint64,uint40,uint256,address,IEigenPod.VALIDATOR_STATUS)
(contracts/pods/EigenPod.sol#361-420):
    External calls:
    -
eigenPodManager.recordOvercommittedBeaconChainETH(podOwner,beaconChainETHStrategyIndex,uin
t256(REQUIRED_BALANCE_GWEI - withdrawalAmountGwei) * GWEI_TO_WEI)
(contracts/pods/EigenPod.sol#382)
    - eigenPodManager.restakeBeaconChainETH(podOwner,REQUIRED_BALANCE_WEI)
(contracts/pods/EigenPod.sol#396)
    - eigenPodManager.restakeBeaconChainETH(podOwner,uint256(withdrawalAmountGwei) *
GWEI_TO_WEI) (contracts/pods/EigenPod.sol#404)
    Event emitted after the call(s):
    - FullWithdrawalRedeemed(validatorIndex,recipient,withdrawalAmountGwei)
(contracts/pods/EigenPod.sol#414)
Reentrancy in EigenPod.stake(bytes,bytes,bytes32) (contracts/pods/EigenPod.sol#158-163):
    External calls:
    - ethPOS.deposit{value:
32000000000000000000}(pubkey,_podWithdrawalCredentials(),signature,depositDataRoot)
(contracts/pods/EigenPod.sol#161)
    Event emitted after the call(s):
    - EigenPodStaked(pubkey) (contracts/pods/EigenPod.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-3
INFO:Detectors:
Address._revert(bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) uses
assembly
    - INLINE ASM
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#236-239)
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#12-80) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#15-77)
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#82-211) uses
assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#83-210)
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#213-270) uses
assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#219-267)
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#272-281) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#276-278)
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#283-292) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#287-289)
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#294-303) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#298-300)
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#305-314) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#309-311)
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#316-325) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#320-322)
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#327-336) uses assembly
```



```
    - INLINE ASM (contracts/libraries/BytesLib.sol#331-333)
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#338-347) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#342-344)
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#349-358) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#353-355)
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#360-369) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#364-366)
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#371-411) uses assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#374-408)
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#413-477) uses
assembly
    - INLINE ASM (contracts/libraries/BytesLib.sol#416-474)
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '>=0.8.0<0.9.0', '^0.8.0', '^0.8.1', '^0.8.2']
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IETHPOSDeposit.sol#12)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - =0.8.12 (contracts/pods/EigenPod.sol#2)
    - =0.8.12 (contracts/pods/EigenPodPausingConstants.sol#2)
    - >=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4)
    - ^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4)
    - ^0.8.2
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
Address._revert(bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) is never
used and should be removed
Address.functionCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never
used and should be removed
Address.functionCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never
used and should be removed
Address.functionCallWithValue(address,bytes,uint256)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never
used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137) is never
used and should be removed
Address.functionDelegateCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#170-172) is never
used and should be removed
Address.functionDelegateCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187) is never
used and should be removed
Address.functionStaticCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#145-147) is never
used and should be removed
Address.functionStaticCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162) is never
used and should be removed
Address.sendValue(address,uint256)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never
used and should be removed
Address.verifyCallResult(bool,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#219-229) is never
used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#195-211) is never
used and should be removed
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5])
(contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21])
(contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3])
(contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8])
(contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BytesLib.concat(bytes,bytes) (contracts/libraries/BytesLib.sol#12-80) is never used and
should be removed
BytesLib.concatStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#82-211) is never
used and should be removed
```

```
BytesLib.equal(bytes,bytes) (contracts/libraries/BytesLib.sol#371-411) is never used and
should be removed
BytesLib.equalStorage(bytes,bytes) (contracts/libraries/BytesLib.sol#413-477) is never
used and should be removed
BytesLib.slice(bytes,uint256,uint256) (contracts/libraries/BytesLib.sol#213-270) is never
used and should be removed
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#272-281) is never used
and should be removed
BytesLib.toBytes32(bytes,uint256) (contracts/libraries/BytesLib.sol#360-369) is never used
and should be removed
BytesLib.toUint128(bytes,uint256) (contracts/libraries/BytesLib.sol#338-347) is never used
and should be removed
BytesLib.toUint16(bytes,uint256) (contracts/libraries/BytesLib.sol#294-303) is never used
and should be removed
BytesLib.toUint256(bytes,uint256) (contracts/libraries/BytesLib.sol#349-358) is never used
and should be removed
BytesLib.toUint32(bytes,uint256) (contracts/libraries/BytesLib.sol#305-314) is never used
and should be removed
BytesLib.toUint64(bytes,uint256) (contracts/libraries/BytesLib.sol#316-325) is never used
and should be removed
BytesLib.toUint8(bytes,uint256) (contracts/libraries/BytesLib.sol#283-292) is never used
and should be removed
BytesLib.toUint96(bytes,uint256) (contracts/libraries/BytesLib.sol#327-336) is never used
and should be removed
Context._msgData()
(contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never
used and should be removed
Initializable._getInitializedVersion()
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#155-
157) is never used and should be removed
Initializable._isInitializing()
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#162-
164) is never used and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
ReentrancyGuard._nonReentrantAfter()
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#64-68)
is never used and should be removed
ReentrancyGuard._nonReentrantBefore()
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#56-62)
is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old
versions
Pragma version^0.8.2
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
allows old versions
```

```
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
allows old versions
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version^0.8.1
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old
versions
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IETHPOSDeposit.sol#12) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version>=0.8.0<0.9.0 (contracts/libraries/BytesLib.sol#9) is too complex
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
Pragma version=0.8.12 (contracts/pods/EigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/pods/EigenPodPausingConstants.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
    - (success) = recipient.call{value: amount}()
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137):
    - (success,returndata) = target.call{value: value}(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#135)
Low level call in Address.functionStaticCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162):
    - (success,returndata) = target.staticcall(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#160)
Low level call in Address.functionDelegateCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187):
    - (success,returndata) = target.delegatecall(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
INFO:Detectors:
Parameter IETHPOSDeposit.deposit(bytes,bytes,bytes,bytes32).withdrawal_credentials
(contracts/interfaces/IETHPOSDeposit.sol#29) is not in mixedCase
Parameter IETHPOSDeposit.deposit(bytes,bytes,bytes,bytes32).deposit_data_root
(contracts/interfaces/IETHPOSDeposit.sol#31) is not in mixedCase
Function IETHPOSDeposit.get_deposit_root() (contracts/interfaces/IETHPOSDeposit.sol#36) is
not in mixedCase
Function IETHPOSDeposit.get_deposit_count() (contracts/interfaces/IETHPOSDeposit.sol#40)
is not in mixedCase
Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not
in mixedCase
Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not
in mixedCase
Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in
CapWords
Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44)
is not in CapWords
Parameter BytesLib.concat(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#12) is
not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#12) is
not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes
(contracts/libraries/BytesLib.sol#82) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes
(contracts/libraries/BytesLib.sol#82) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes
(contracts/libraries/BytesLib.sol#213) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start
(contracts/libraries/BytesLib.sol#213) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length
(contracts/libraries/BytesLib.sol#213) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#272)
is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (contracts/libraries/BytesLib.sol#272)
is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#283) is
not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (contracts/libraries/BytesLib.sol#283) is
not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#294)
is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (contracts/libraries/BytesLib.sol#294)
is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#305)
is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#305)
is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#316)
is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (contracts/libraries/BytesLib.sol#316)
is not in mixedCase
```

```
Parameter BytesLib.toUint96(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#327)
is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (contracts/libraries/BytesLib.sol#327)
is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#338)
is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (contracts/libraries/BytesLib.sol#338)
is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#349)
is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (contracts/libraries/BytesLib.sol#349)
is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (contracts/libraries/BytesLib.sol#360)
is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (contracts/libraries/BytesLib.sol#360)
is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (contracts/libraries/BytesLib.sol#371) is
not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (contracts/libraries/BytesLib.sol#371) is
not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes
(contracts/libraries/BytesLib.sol#413) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes
(contracts/libraries/BytesLib.sol#413) is not in mixedCase
Parameter EigenPod.initialize(address)._podOwner (contracts/pods/EigenPod.sol#152) is not
in mixedCase
Variable EigenPod.REQUIRED_BALANCE_GWEI (contracts/pods/EigenPod.sol#53) is not in
mixedCase
Variable EigenPod.REQUIRED_BALANCE_WEI (contracts/pods/EigenPod.sol#56) is not in
mixedCase
Variable EigenPod.__gap (contracts/pods/EigenPod.sol#473) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
Variable EigenPod.REQUIRED_BALANCE_GWEI (contracts/pods/EigenPod.sol#53) is too similar to
EigenPod.constructor(IETHPOSDeposit,IDelayedWithdrawalRouter,IEigenPodManager,uint256)._RE
QUIRED_BALANCE_WEI (contracts/pods/EigenPod.sol#140)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-
too-similar
INFO:Detectors:
BytesLib.toAddress(bytes,uint256) (contracts/libraries/BytesLib.sol#272-281) uses literals
with too many digits:
    - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000
(contracts/libraries/BytesLib.sol#277)
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
```

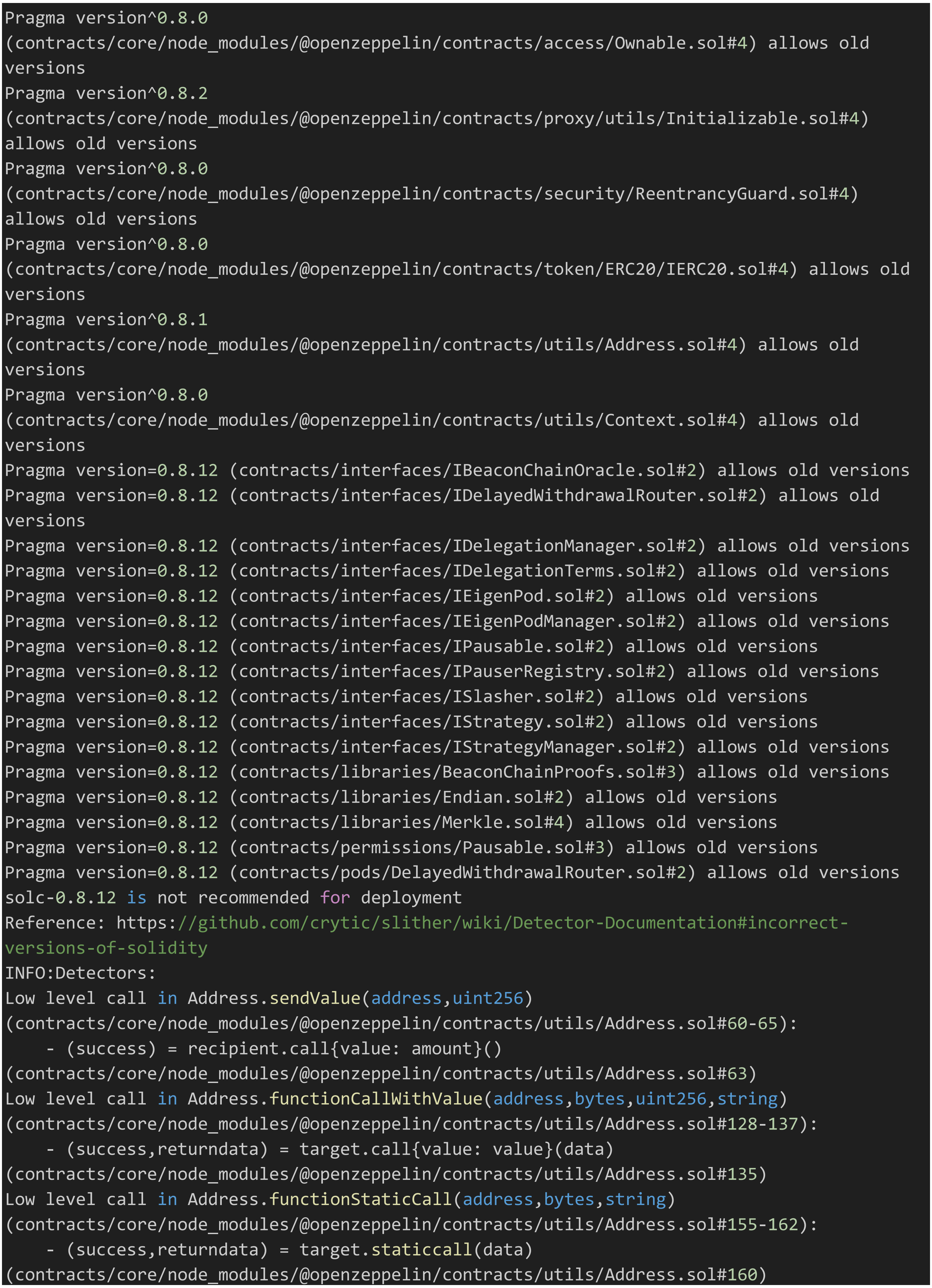
```
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS
(contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS
(contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS
(contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX
(contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX
(contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX
(contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX
(contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX
(contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX
(contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX
(contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
EigenPodPausingConstants.PAUSED_NEW_EIGENPODS
(contracts/pods/EigenPodPausingConstants.sol#10) is never used in EigenPod
(contracts/pods/EigenPod.sol#34-475)
EigenPodPausingConstants.PAUSED_WITHDRAW_RESTAKED_ETH
(contracts/pods/EigenPodPausingConstants.sol#12) is never used in EigenPod
(contracts/pods/EigenPod.sol#34-475)
EigenPod.__gap (contracts/pods/EigenPod.sol#473) is never used in EigenPod
(contracts/pods/EigenPod.sol#34-475)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable
INFO:Slither:contracts/pods/EigenPod.sol analyzed (24 contracts with 85 detectors), 165
result(s) found
```

21.DelayerdWIthdrwalRouter.sol

```
INFO:Detectors:
Address._revert(bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) uses
assembly
    - INLINE ASM
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#236-239)
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity are used:
    - Version used: ['=0.8.12', '^0.8.0', '^0.8.1', '^0.8.2']
    - =0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2)
    - =0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationManager.sol#2)
    - =0.8.12 (contracts/interfaces/IDelegationTerms.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPod.sol#2)
    - =0.8.12 (contracts/interfaces/IEigenPodManager.sol#2)
    - =0.8.12 (contracts/interfaces/IPausable.sol#2)
    - =0.8.12 (contracts/interfaces/IPauserRegistry.sol#2)
    - =0.8.12 (contracts/interfaces/ISlasher.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategy.sol#2)
    - =0.8.12 (contracts/interfaces/IStrategyManager.sol#2)
    - =0.8.12 (contracts/libraries/BeaconChainProofs.sol#3)
    - =0.8.12 (contracts/libraries/Endian.sol#2)
    - =0.8.12 (contracts/libraries/Merkle.sol#4)
    - =0.8.12 (contracts/permissions/Pausable.sol#3)
    - =0.8.12 (contracts/pods/DelayedWithdrawalRouter.sol#2)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
    - ^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
    - ^0.8.0 (contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4)
    - ^0.8.1 (contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4)
    - ^0.8.2
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-
directives-are-used
INFO:Detectors:
Address._revert(bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#231-243) is never
used and should be removed
Address.functionCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never
used and should be removed
Address.functionCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never
used and should be removed
Address.functionCallWithValue(address,bytes,uint256)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never
used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137) is never
used and should be removed
Address.functionDelegateCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#170-172) is never
used and should be removed
Address.functionDelegateCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187) is never
used and should be removed
Address.functionStaticCall(address,bytes)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#145-147) is never
used and should be removed
```

```
Address.functionStaticCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162) is never
used and should be removed
Address.verifyCallResult(bool,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#219-229) is never
used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#195-211) is never
used and should be removed
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5])
(contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21])
(contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3])
(contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8])
(contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32)
(contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32)
(contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[])
(contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes3
2[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be
removed
Context._msgData()
(contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never
used and should be removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used
and should be removed
Initializable._disableInitializers()
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#144-
150) is never used and should be removed
Initializable._getInitializedVersion()
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#155-
157) is never used and should be removed
Initializable._isInitializing()
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#162-
164) is never used and should be removed
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used
and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
```



```
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old
versions
Pragma version^0.8.2
(contracts/core/node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
allows old versions
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
allows old versions
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old
versions
Pragma version^0.8.1
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old
versions
Pragma version^0.8.0
(contracts/core/node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IBeaconChainOracle.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelayedWithdrawalRouter.sol#2) allows old
versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IDelegationTerms.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPod.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IEigenPodManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPausable.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IPauserRegistry.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/ISlasher.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategy.sol#2) allows old versions
Pragma version=0.8.12 (contracts/interfaces/IStrategyManager.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
Pragma version=0.8.12 (contracts/permissions/Pausable.sol#3) allows old versions
Pragma version=0.8.12 (contracts/pods/DelayedWithdrawalRouter.sol#2) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
    - (success) = recipient.call{value: amount}()
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#128-137):
    - (success,returndata) = target.call{value: value}(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#135)
Low level call in Address.functionStaticCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#155-162):
    - (success,returndata) = target.staticcall(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#160)
```

```
Low level call in Address.functionDelegateCall(address,bytes,string)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#180-187):
    - (success,returndata) = target.delegatecall(data)
(contracts/core/node_modules/@openzeppelin/contracts/utils/Address.sol#185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IEigenPod.REQUIRED_BALANCE_GWEI() (contracts/interfaces/IEigenPod.sol#47) is not
in mixedCase
Function IEigenPod.REQUIRED_BALANCE_WEI() (contracts/interfaces/IEigenPod.sol#50) is not
in mixedCase
Enum IEigenPod.VALIDATOR_STATUS (contracts/interfaces/IEigenPod.sol#22-27) is not in
CapWords
Enum IEigenPod.PARTIAL_WITHDRAWAL_CLAIM_STATUS (contracts/interfaces/IEigenPod.sol#40-44)
is not in CapWords
Variable Pausable.__gap (contracts/permissions/Pausable.sol#115) is not in mixedCase
Parameter
DelayedWithdrawalRouter.initialize(address,IPauserRegistry,uint256,uint256)._pauserRegistr
y (contracts/pods/DelayedWithdrawalRouter.sol#49) is not in mixedCase
Parameter
DelayedWithdrawalRouter.initialize(address,IPauserRegistry,uint256,uint256)._withdrawalDel
ayBlocks (contracts/pods/DelayedWithdrawalRouter.sol#49) is not in mixedCase
Variable DelayedWithdrawalRouter._userWithdrawals
(contracts/pods/DelayedWithdrawalRouter.sol#30) is not in mixedCase
Variable DelayedWithdrawalRouter.__gap (contracts/pods/DelayedWithdrawalRouter.sol#177) is
not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS
(contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS
(contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS
(contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX
(contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX
(contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX
(contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX
(contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX
(contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX
(contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX
(contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
Pausable.UNPAUSE_ALL (contracts/permissions/Pausable.sol#22) is never used in
DelayedWithdrawalRouter (contracts/pods/DelayedWithdrawalRouter.sol#11-179)
Pausable.PAUSE_ALL (contracts/permissions/Pausable.sol#23) is never used in
DelayedWithdrawalRouter (contracts/pods/DelayedWithdrawalRouter.sol#11-179)
```

```
DelayedWithdrawalRouter.__gap (contracts/pods/DelayedWithdrawalRouter.sol#177) is never
used in DelayedWithdrawalRouter (contracts/pods/DelayedWithdrawalRouter.sol#11-179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable
INFO:Slither:contracts/pods/DelayedWithdrawalRouter.sol analyzed (22 contracts with 85
detectors), 98 result(s) found
```

22.BeaconChainProofs.sol

```
INFO:Detectors:
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#53-58)
    - INLINE ASM (contracts/libraries/Merkle.sol#61-66)
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) uses assembly
    - INLINE ASM (contracts/libraries/Merkle.sol#104-109)
    - INLINE ASM (contracts/libraries/Merkle.sol#112-117)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
BeaconChainProofs.computePhase0BeaconBlockHeaderRoot(bytes32[5])
(contracts/libraries/BeaconChainProofs.sol#130-138) is never used and should be removed
BeaconChainProofs.computePhase0BeaconStateRoot(bytes32[21])
(contracts/libraries/BeaconChainProofs.sol#140-148) is never used and should be removed
BeaconChainProofs.computePhase0Eth1DataRoot(bytes32[3])
(contracts/libraries/BeaconChainProofs.sol#160-168) is never used and should be removed
BeaconChainProofs.computePhase0ValidatorRoot(bytes32[8])
(contracts/libraries/BeaconChainProofs.sol#150-158) is never used and should be removed
BeaconChainProofs.getBalanceFromBalanceRoot(uint40,bytes32)
(contracts/libraries/BeaconChainProofs.sol#178-183) is never used and should be removed
BeaconChainProofs.verifyValidatorBalance(uint40,bytes32,bytes,bytes32)
(contracts/libraries/BeaconChainProofs.sol#221-237) is never used and should be removed
BeaconChainProofs.verifyValidatorFields(uint40,bytes32,bytes,bytes32[])
(contracts/libraries/BeaconChainProofs.sol#192-212) is never used and should be removed
BeaconChainProofs.verifyWithdrawalProofs(bytes32,BeaconChainProofs.WithdrawalProofs,bytes3
2[]) (contracts/libraries/BeaconChainProofs.sol#245-295) is never used and should be
removed
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) is never used
and should be removed
Merkle.merkleizeSha256(bytes32[]) (contracts/libraries/Merkle.sol#129-153) is never used
and should be removed
Merkle.processInclusionProofKeccak(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#48-70) is never used and should be removed
Merkle.processInclusionProofSha256(bytes,bytes32,uint256)
(contracts/libraries/Merkle.sol#99-121) is never used and should be removed
Merkle.verifyInclusionKeccak(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#29-36) is never used and should be removed
```

```
Merkle.verifyInclusionSha256(bytes,bytes32,bytes32,uint256)
(contracts/libraries/Merkle.sol#80-87) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version=0.8.12 (contracts/libraries/BeaconChainProofs.sol#3) allows old versions
Pragma version=0.8.12 (contracts/libraries/Endian.sol#2) allows old versions
Pragma version=0.8.12 (contracts/libraries/Merkle.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity
INFO:Detectors:
Endian.fromLittleEndianUint64(bytes32) (contracts/libraries/Endian.sol#5-19) uses literals
with too many digits:
    - (n >> 56) | ((0x00FF000000000000 & n) >> 40) | ((0x0000FF0000000000 & n) >> 24) |
((0x000000FF00000000 & n) >> 8) | ((0x00000000FF000000 & n) << 8) | ((0x0000000000FF0000 &
n) << 24) | ((0x000000000000FF00 & n) << 40) | ((0x00000000000000FF & n) << 56)
(contracts/libraries/Endian.sol#10-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BeaconChainProofs.NUM_BEACON_BLOCK_BODY_FIELDS
(contracts/libraries/BeaconChainProofs.sol#17) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_HEADER_FIELDS
(contracts/libraries/BeaconChainProofs.sol#29) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_EXECUTION_PAYLOAD_FIELDS
(contracts/libraries/BeaconChainProofs.sol#33) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.EXECUTION_PAYLOAD_FIELD_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#34) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#38) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_TREE_HEIGHT
(contracts/libraries/BeaconChainProofs.sol#41) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_TREE_HEIGHT (contracts/libraries/BeaconChainProofs.sol#44)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.NUM_WITHDRAWAL_FIELDS (contracts/libraries/BeaconChainProofs.sol#48) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#63) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.PROPOSER_INDEX_INDEX (contracts/libraries/BeaconChainProofs.sol#64) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.STATE_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#68) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_ROOTS_INDEX (contracts/libraries/BeaconChainProofs.sol#70) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.ETH_1_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#71) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
```

```
BeaconChainProofs.EXECUTION_PAYLOAD_HEADER_INDEX
(contracts/libraries/BeaconChainProofs.sol#74) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICAL_BATCH_STATE_ROOT_INDEX
(contracts/libraries/BeaconChainProofs.sol#75) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWAL_CREDENTIALS_INDEX
(contracts/libraries/BeaconChainProofs.sol#78) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_BALANCE_INDEX (contracts/libraries/BeaconChainProofs.sol#79)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_SLASHED_INDEX (contracts/libraries/BeaconChainProofs.sol#80)
is never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.VALIDATOR_WITHDRAWABLE_EPOCH_INDEX
(contracts/libraries/BeaconChainProofs.sol#81) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWALS_ROOT_INDEX (contracts/libraries/BeaconChainProofs.sol#85) is
never used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_INDEX_INDEX
(contracts/libraries/BeaconChainProofs.sol#91) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.WITHDRAWAL_VALIDATOR_AMOUNT_INDEX
(contracts/libraries/BeaconChainProofs.sol#92) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.HISTORICALBATCH_STATEROOTS_INDEX
(contracts/libraries/BeaconChainProofs.sol#95) is never used in BeaconChainProofs
(contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.SLOTS_PER_EPOCH (contracts/libraries/BeaconChainProofs.sol#98) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
BeaconChainProofs.UINT64_MASK (contracts/libraries/BeaconChainProofs.sol#100) is never
used in BeaconChainProofs (contracts/libraries/BeaconChainProofs.sol#12-298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable
INFO:Slither:contracts/libraries/BeaconChainProofs.sol analyzed (3 contracts with 85
detectors), 46 result(s) found
```

References

[1] "Binance," [Online]. Available: https://academy.binance.com/en/articles/what-are-smart-contracts?utm_source=googleadwords_int_pmax&utm_medium=cpc&ref=HDYAHEES&gclid=CjwKCAjwvJyjBhApEiwAWz2nLdzwP0JL6BvAfLJL4qsvm0N98t4SYoRrqMXrG664o4ONoGHYROErshoC9x8QAvD_BwE. [Accessed 27 04 2023].

[2] "Invetopedia," [Online]. Available: https://www.investopedia.com/terms/s/smart-contracts.asp.

[3] "Cqinspect," [Online]. Available: https://www.coinspect.com/smart-contract-audit/?gclid=CjwKCAjwvJyjBhApEiwAWz2nLfYH_PY26p4aDfjC12ee04DDfnVXUJwgfa5zgXK5tGW1Cun2y_KxiRoCfqgQAvD_BwE.

[4] "chain link," [Online]. Available: https://blog.chain.link/how-to-audit-smart-contract/.

[5] "Dev team.spca," [Online]. Available: https://www.devteam.space/blog/how-to-audit-a-smart-contract-a-guide/.

[6] "hckermoon," [Online]. Available: https://hackernoon.com/hack-solidity-reentrancy-attack.

[7] "secure coding," [Online]. Available: https://www.securecoding.com/blog/integer-overflow-attack-and-prevention/.

[8] "finxter," [Online]. Available: https://blog.finxter.com/denial-of-service-dos-attack-on-smart-contracts/.

[9] "fyeo," [Online]. Available: https://www.fyeo.io/post/denial-of-service-block-gas-limit.

[10] "investopidia," [Online]. Available: https://www.investopedia.com/terms/f/frontrunning.asp.