QuillAudits

Audit Report
May, 2022

For

STAKEALL

Table of Content

Executive Summary

Project Name	Staking Shuttle
Timeline	20th April, 2022 to 20th May, 2022
Method	Manual Review, Functional Testing, Automated Testing etc.
Scope of Audit	The scope of this audit was to analyse Staking Shuttle codebase for quality, security, and correctness.
Git Repo link	*https://github.com/stakeall/cross-chain-staking-shuttle*
Commit Hash	2524a365a71beca88db604058f42dd1feb224e16
Fixed In	1. d7202d1d46bbef881687262a9027f820cda944ca 2. c0dc70ab72aa15e1ce9cf72f954dc9109bf4c1af 3. 1916d2df760df031664f15e80581801ed4aff496

High
Medium
Low
Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	2
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	1	1	8

Types of Severities

High

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Checked Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- Exception Disorder
- Gasless Send
- Use of tx.origin
- Compiler version not fixed
- Address hardcoded
- Divide before multiply
- Integer overflow/underflow
- Dangerous strict equalities
- Tautology or contradiction
- Return values of low-level calls
- Missing Zero Address Validation
- Private modifier
- Revert/require functions
- Using block.timestamp
- Multiple Sends
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Manual Testing

High Severity Issues

No issues found

Medium Severity Issues

1. Parameter and msg.value mismatch

Line #93 function deposit() external payable

Contract: ChildPool.sol

Description

When passing amount to deposit function, it often mismatches with the matic amount sent to due to the gas price we pay.

Remediation

Instead of taking amount as a parameter, set the value of amount from msg.value. This will ensure correctness and will also allow us to remove a require check hence saving some gas.

Status

Fixed

Low Severity Issues

2. Missing setters

Contract: ChildPool.sol

Description

The contract initialises values for fundCollector and feeBeneficiary, which might need updating in the future. It can be because of an upgrade or because of them being compromised on security.

Recommendation

We should add setters for fundCollector and feeBeneficiary, which can be called by an authorized user.

Status

Fixed

Informational Issues

3. Confusing error messages

Recommendation

The error messages in the code base should be simple, clear and straightforward.

Status

Acknowledged

4. Unnecessary initialization

Contract: ChildPool.sol

Description

In solidity variables which are not initialized are set to zero by default if they are of the type uint256, but in initialization of ChildPool, we are setting this variables to zero explicitly.

```
currentShuttle = 0;
enroutedShuttle = 0;
availableMaticBalance = 0;
availableStMaticBalance = 0;
```

Recommendation

Remove the redundant initialization to save on gas.

Status

Fixed

5. Unnecessary check

Contract: ChildPool.sol

Line #95

```
require(
    shuttles[currentShuttle].status == ShuttleStatus.AVAILABLE,
    "!Shuttle"
);
```

Description

There is an unnecessary required check in deposit

Recommendation

Remove the require check since there will always be a current shuttle which will be available.

Status

Fixed

6. Method can be made external

Contract: FxStateChildTunnel.sol, FxStateRootTunnel.sol

Line #36, 27 function readData() public view returns

Description

There is an unnecessary required check in deposit

Recommendation

Change the visibility to external to reduce gas costs of operation.

Status

Fixed

7. Repeated code lines

Contract: RootPool.sol

Description

The methods crossChainStake and cancelShuttle share a lot of common logic, which why there are a same bunch of lines written in both methods which can be easily eliminated.

Recommendation

Create an internal method and move the common code into it, and call this method from both crossChainStake and cancelShuttle.

Status

Fixed

8. Incorrect parameter description

Contract: RootPool.sol

Line #24

```
* @param _rootTunnel - Address of the child tunnel.
* @param _erc20PredicateProxy - Address of the owner
* @param _polidoAdapter - Address of the owner
* @param _maticToken - Address of the owner
```

Description

The netspec comments added for parameter description are incorrect

Recommendation

Add correct description for the comments

Status

Fixed

9. Unused variable

Contract: RootPool.sol

Line #16 address public erc20PredicateProxy;

Description

The address is set for erc20PredicateProxy but never used.

Recommendation

Remove unused variables

Status

Fixed

10. Internal method naming convention

Recommendation

Internal method names should be preceded by '_' to differentiate them visibly from external and public methods according to solidity naming convention.

Status

Fixed

11. Missing License Identifier

Contract: RootPool.sol

Recommendation

Add SPDX-License-Identifier for all the contracts

Status

Fixed

12. Missing netspec comments

Recommendation

We recommend adding netspec comments for each method and variables for better readability and understanding of code.

Status

Acknowledged

Functional Testing

Contracts

L2 (MUMBAI)

- FxChild 0xCf73231F28B7331BBe3124B907840A94851f9f11
- FxStateChildTunnel 0x0e967b0BCCAB110F462DfA6420266A1A6B42813D
- FundsCollector 0x3b01704DDD6f3115734D1E7276cEdA57A7F87765
- _maticToken 0x0000000000000000000000000000000000001010
- _stMaticToken 0xa337f0B897a874DE1E9F75944629a03F911cFbE8
- ChildPoolProxy 0xACDA977fa970521b5be476A47b39B8E29C08B021

L1 (GOERLI)

- CheckpointManager 0x2890bA17EfE978480615e330ecB65333b880928e
- FxRoot 0x3d1d3E34f7fB6D26245E6640E1c50710eFFf15bA
- FxStateRootTunnel 0x0e967b0BCCAB110F462DfA6420266A1A6B42813D
- WithdrawManagerProxy 0x2923C8dD6Cdf6b2507ef91de74F1d5E0F11Eac53
- erc20PredicateBurnOnly 0xf213e8fF5d797ed2B052D3b96C11ac71dB358027,
- depositManagerProxy 0x7850ec290A2e2F40B82Ed962eaf30591bb5f5C96',
- erc20PredicateProxy 0xdD6596F2029e6233DEFfaCa316e6A95217d4Dc34,
- poLidoAdapter 0xf8bb8087F9967Edf6B0D26D146fA978A953EC2A5,
- maticToken 0x499d11e0b6eac7c0593d8fb292dcbbf815fb29ae,
- childPoolFundCollector 0x3b01704DDD6f3115734D1E7276cEdA57A7F87765,
- RootPoolProxy 0xACDA977fa970521b5be476A47b39B8E29C08B021

Transactions

StateRootTunnel

setFxChildTunnel
0x575c114de96c777f2875583a6ca6536d4c906c712e7309c62463e8b8d4a1ae9d

setPool
0xde95324854010374b4bf41663191d8b86f0abb495ecfa662bf6b94f9031c76db

StateChildTunnel

setFxRootTunnel
0x05e28cf9391cb187e071fdce81dfe296c0fad37659f8334f7aeb291bbe1881bb

setPool
0x78566520494904470b36f62b050206a66a1d3692cd608aa86d347b81fa47c6bf

FundsCollector

setChildPool
0x915b28290c0a747a47ca8aee185a7550807ffe5a50e38f4678f8f07138490957

ChildPool

Deposit
0x2aaa4b69be9285880380e5db9bfcb3d8997a82881fec060524a9867a1cb19132
0xe083a4912cb62a9311559cf7d1949b8fefc22065c4d59fd117b93af98bbcd723
0x84de6b9aace0650dbe993f969e0ec629ad455d6c07855c82195cdf1eeba53fe9

EnrouteShuttle
0xddb6ea5d94f4fa4fee18cd0d8e80fea0922c1300d59df6799719fe30bcf70cd9
0xf58e927ee7051adb14d745a24167e96d37229588188fa84688d2e8f8269bce6f

Claim
0xd7fc0e2996021b42930c7599e80edd8aac7046923c1abfef4945a248a8f7db2c

setFee
0x359b0e4d7c9a0dd13f822aa2f79b8d2ac089fb35f315ea9cc52e3682450a6bbc

setShuttleExpiry
0x38eb871ee76b7190b63c633fe3fbfef2b8af8f054a61244a92b6d0d0f6b7cf72

cancelShuttle
0x5ab0f7eeb7215ca4c42c11efb349918b1e57415aff09d1c50e4507aa3888e3c6

claim(cancelled)
0x3e7aafb3aefd056675b98b6f85f396a834f4e3bca7577b28b9554feb37e036c7

expireShuttle
0x9521bba9998969c080caba65679a5e685ee09d35ab14d06dadf42ffc53bba39b

claim(expired)
0xe5f0574c64b9caf3e3e0922f3b2cd87f21049bb98e5791054b5407653b78a260

pause
0x6f1b365102945b484ca2c1d06e781ddd6359a91a6021de1e35ab7db1f18cfbd1

arrival(paused)
0x70be466f42d7f615ba27adf0c1708b0b6727bebbb0169a99265ae3cb40d69304

unpause
0xe63bed3cb862b43ec374fa8b0fae8a743f1525b07087223f62216b5509b45260

arrival(cancelled on Root)
0x5d0893bed3c9e92eea53f7aab706e2e5a5c56ec8a31682139b850dc1bc2c86cd

claim(cancelled on Root)
0x29d6efe7e6b848b5747d93dd835a0517ebe3c3110bbe2fd1dfae50ed2592566c

RootPool

startExitWithBurntTokens
0x2ee490ec04dec20f75b65155295df2ac4ceccf58c247a5a5b132ae983d134268
0xaeb8227eb25c2db45c27e31558900ad219f02cd59991af40851559591646d1fc

crossChainStake
0x40edb61107f29a786480cb8ff9f6900278ef375e5c0b38ebda094e556b3e6a3c

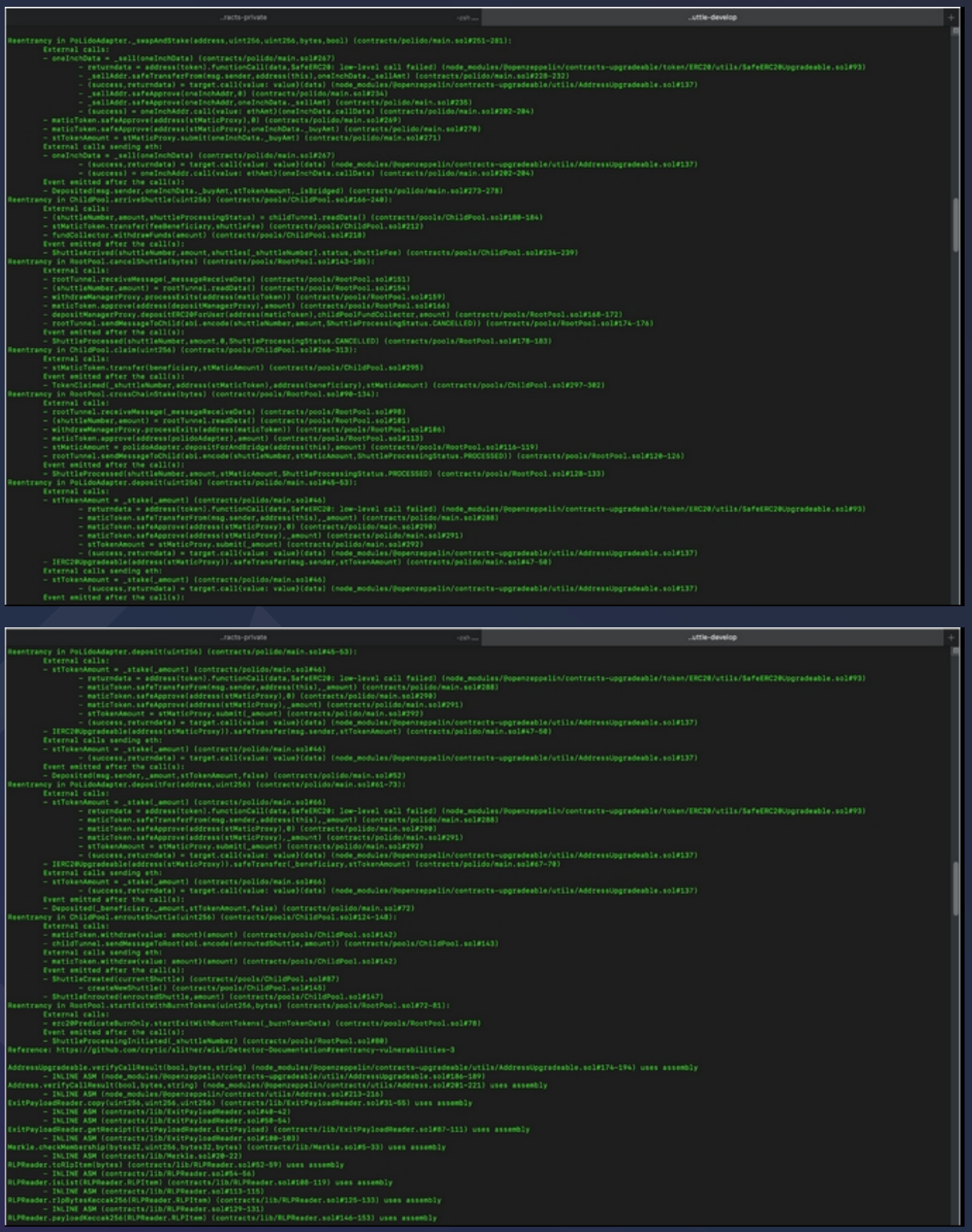
cancelShuttle
0xbbe89297c8ba3b7b66bba9ec084b01706c782d9fc91b81bcabc191854239a651

Automated Tests

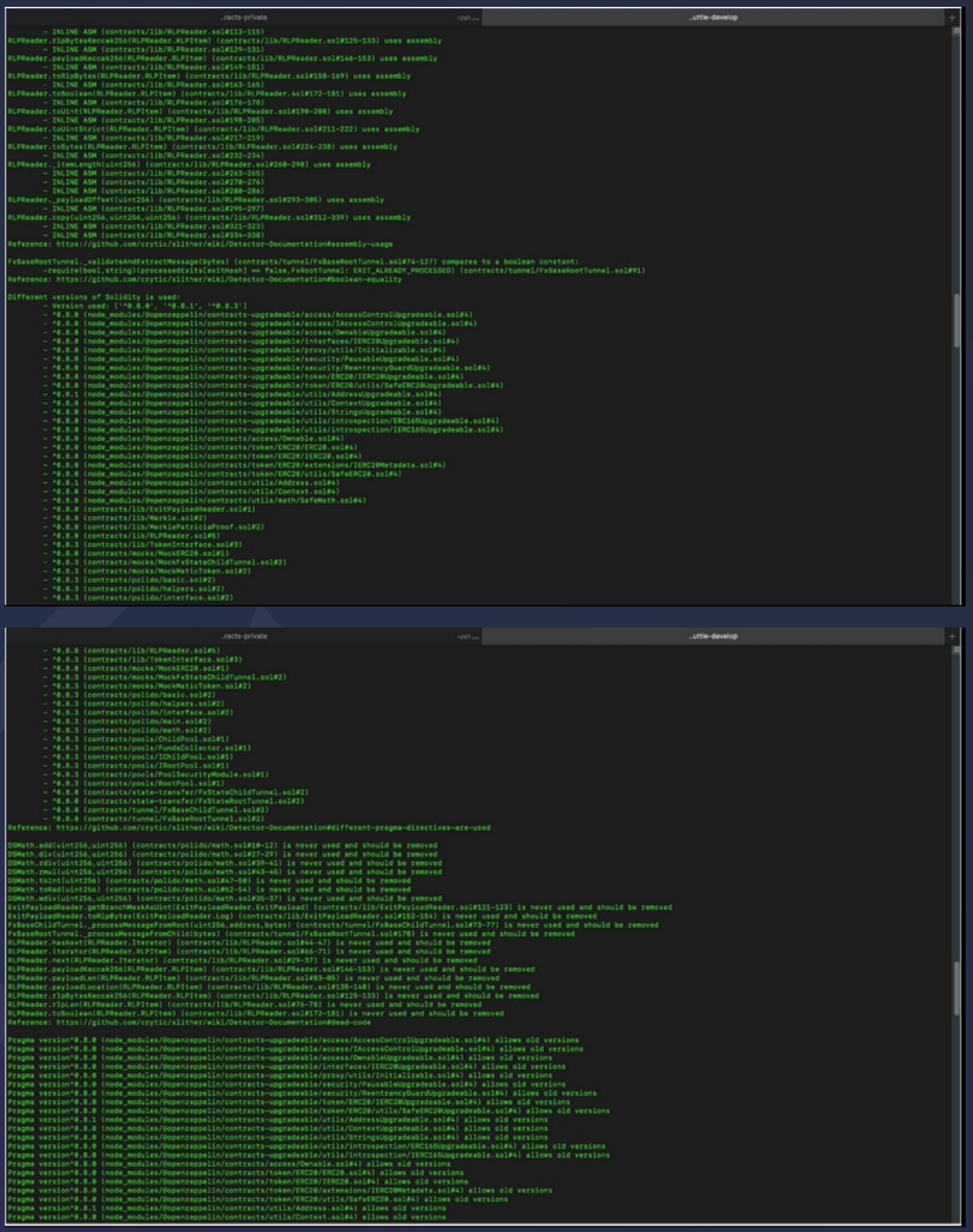
Slither

```
ChildPool.enrouteShuttle(uint256) (contracts/pools/ChildPool.sol#124-148) sends eth to arbitrary user
        Dangerous calls:
        - maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#functions-that-send-ether-to-arbitrary-destinations

Reentrancy in ChildPool.enrouteShuttle(uint256) (contracts/pools/ChildPool.sol#124-148):
        External calls:
        - maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
        - childTunnel.sendMessageToRoot(abi.encode(enroutedShuttle,amount)) (contracts/pools/ChildPool.sol#143)
        External calls sending eth:
        - maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
        State variables written after the call(s):
        - createNewShuttle() (contracts/pools/ChildPool.sol#145)
                - shuttles[currentShuttle] = Shuttle(0,ShuttleStatus.AVAILABLE,0,block.number.add(shuttleExpiry)) (contracts/pools/ChildPool.sol#81-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#235) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240) ignores return value by stMaticToken.transfer(feeBeneficiary,shuttleFee) (contracts/pools/ChildPool.sol#212)
ChildPool.claim(uint256) (contracts/pools/ChildPool.sol#266-313) ignores return value by stMaticToken.transfer(beneficiary,stMaticAmount) (contracts/pools/ChildPool.sol#295)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

Contract locking ether found:
        Contract MockERC20 (contracts/mocks/MockERC20.sol#6-40) has payable functions:
         - MockERC20.constructor(string,string,address,uint256) (contracts/mocks/MockERC20.sol#7-14)
        But does not have a function to withdraw the ether
Contract locking ether found:
        Contract MockMaticToken (contracts/mocks/MockMaticToken.sol#4-12) has payable functions:
         - MockMaticToken.withdraw(uint256) (contracts/mocks/MockMaticToken.sol#8-10)
        But does not have a function to withdraw the ether
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#contracts-that-lock-ether

Reentrancy in ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240):
        External calls:
        - (shuttleNumber,amount,shuttleProcessingStatus) = childTunnel.readData() (contracts/pools/ChildPool.sol#180-184)
        State variables written after the call(s):
        - enroutedShuttle = 0 (contracts/pools/ChildPool.sol#193)
        - shuttles[_shuttleNumber].recievedToken = recievedToken (contracts/pools/ChildPool.sol#209)
        - shuttles[_shuttleNumber].status = ShuttleStatus.ARRIVED (contracts/pools/ChildPool.sol#210)
Reentrancy in ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240):
        External calls:
        - (shuttleNumber,amount,shuttleProcessingStatus) = childTunnel.readData() (contracts/pools/ChildPool.sol#180-184)
        - fundCollector.withdrawFunds(amount) (contracts/pools/ChildPool.sol#218)
        State variables written after the call(s):
        - shuttles[_shuttleNumber].recievedToken = 0 (contracts/pools/ChildPool.sol#230)
        - shuttles[_shuttleNumber].status = ShuttleStatus.CANCELLED (contracts/pools/ChildPool.sol#231)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

PoLidoAdapter._sell(OneInchData).ethAmt (contracts/polido/main.sol#224) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

RootPool.crossChainStake(bytes) (contracts/pools/RootPool.sol#90-134) ignores return value by maticToken.approve(address(polidoAdapter),amount) (contracts/pools/RootPool.sol#113)
RootPool.cancelShuttle(bytes) (contracts/pools/RootPool.sol#143-185) ignores return value by maticToken.approve(address(depositManagerProxy),amount) (contracts/pools/RootPool.sol#166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

MockERC20.constructor(string,string,address,uint256).name (contracts/mocks/MockERC20.sol#8) shadows:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#63-64) (function)
```

```
MockERC20.constructor(string,string,address,uint256).name (contracts/mocks/MockERC20.sol#8) shadows:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64) (function)
        - IERC20Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#17) (function)
MockERC20.constructor(string,string,address,uint256).symbol (contracts/mocks/MockERC20.sol#9) shadows:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72) (function)
        - IERC20Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

FundsCollector.setChildPool(address) (contracts/pools/FundsCollector.sol#43-45) should emit an event for:
        - childPool = _childPool (contracts/pools/FundsCollector.sol#44)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control

PoLidoAdapter.changeFeePercentage(uint256) (contracts/polido/main.sol#35-37) should emit an event for:
        - feePercentage = _newFeePercentage (contracts/polido/main.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._feeBeneficiary (contracts/pools/ChildPool.sol#60) lacks a zero-check on :
                - feeBeneficiary = _feeBeneficiary (contracts/pools/ChildPool.sol#63)
ChildPool.claim(uint256).beneficiary (contracts/pools/ChildPool.sol#280) lacks a zero-check on :
                - beneficiary.transfer(balance) (contracts/pools/ChildPool.sol#305)
FundsCollector.setChildPool(address)._childPool (contracts/pools/FundsCollector.sol#43) lacks a zero-check on :
                - childPool = _childPool (contracts/pools/FundsCollector.sol#44)
RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._erc20PredicateProxy (contracts/pools/RootPool.sol#39) lack
s a zero-check on :
                - erc20PredicateProxy = _erc20PredicateProxy (contracts/pools/RootPool.sol#53)
RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._childPoolFundCollector (contracts/pools/RootPool.sol#42) l
acks a zero-check on :
                - childPoolFundCollector = _childPoolFundCollector (contracts/pools/RootPool.sol#56)
FxBaseChildTunnel.setFxRootTunnel(address)._fxRootTunnel (contracts/tunnel/FxBaseChildTunnel.sol#37) lacks a zero-check on :
                - fxRootTunnel = _fxRootTunnel (contracts/tunnel/FxBaseChildTunnel.sol#39)
FxStateChildTunnel.setPool(address)._pool (contracts/state-transfer/FxStateChildTunnel.sol#45) lacks a zero-check on :
                - pool = _pool (contracts/state-transfer/FxStateChildTunnel.sol#46)
FxBaseRootTunnel.setFxChildTunnel(address)._fxChildTunnel (contracts/tunnel/FxBaseRootTunnel.sol#57) lacks a zero-check on :
                - fxChildTunnel = _fxChildTunnel (contracts/tunnel/FxBaseRootTunnel.sol#59)
FxStateRootTunnel.setPool(address)._pool (contracts/state-transfer/FxStateRootTunnel.sol#36) lacks a zero-check on :
                - pool = _pool (contracts/state-transfer/FxStateRootTunnel.sol#37)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240):
        External calls:
        - (shuttleNumber,amount,shuttleProcessingStatus) = childTunnel.readData() (contracts/pools/ChildPool.sol#180-184)
        State variables written after the call(s):
        - availableStMaticBalance = availableStMaticBalance.add(recievedToken) (contracts/pools/ChildPool.sol#205-207)
Reentrancy in ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240):
        External calls:
        - (shuttleNumber,amount,shuttleProcessingStatus) = childTunnel.readData() (contracts/pools/ChildPool.sol#180-184)
        - fundCollector.withdrawFunds(amount) (contracts/pools/ChildPool.sol#218)
        State variables written after the call(s):
        - availableMaticBalance = availableMaticBalance.add(amount) (contracts/pools/ChildPool.sol#229)
Reentrancy in ChildPool.enrouteShuttle(uint256) (contracts/pools/ChildPool.sol#124-148):
        External calls:
        - maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
        - childTunnel.sendMessageToRoot(abi.encode(enroutedShuttle,amount)) (contracts/pools/ChildPool.sol#143)
        External calls sending eth:
        - maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
        State variables written after the call(s):
        - createNewShuttle() (contracts/pools/ChildPool.sol#145)
                - currentShuttle = currentShuttle.add(1) (contracts/pools/ChildPool.sol#80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in PoLidoAdapter._swapAndStake(address,uint256,uint256,bytes,bool) (contracts/polido/main.sol#251-281):
        External calls:
        - oneInchData = _sell(oneInchData) (contracts/polido/main.sol#267)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#93)
```


..racts-private
..utils-develop
Reentrancy in PoLidoAdapter._swapAndStake(address,uint256,uint256,bytes,bool) (contracts/polido/main.sol#251-281):
External calls:
- oneInchData = _sell(oneInchData) (contracts/polido/main.sol#267)
- returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#93)
- maticToken.safeTransferFrom(msg.sender,address(this),_sellAmt) (contracts/polido/main.sol#228-232)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
- _sellAddr.safeApprove(oneInchAddr,0) (contracts/polido/main.sol#234)
- _sellAddr.safeApprove(oneInchAddr,oneInchData._sellAmt) (contracts/polido/main.sol#235)
- (success) = oneInchAddr.call{value: ethAmt}(oneInchData.callData) (contracts/polido/main.sol#202-204)
- maticToken.safeApprove(address(stMaticProxy),0) (contracts/polido/main.sol#269)
- maticToken.safeApprove(address(stMaticProxy),oneInchData._buyAmt) (contracts/polido/main.sol#270)
- stTokenAmount = stMaticProxy.submit(oneInchData._buyAmt) (contracts/polido/main.sol#271)
External calls sending eth:
- oneInchData = _sell(oneInchData) (contracts/polido/main.sol#267)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
- (success) = oneInchAddr.call{value: ethAmt}(oneInchData.callData) (contracts/polido/main.sol#202-204)
Event emitted after the call(s):
- Deposited(msg.sender,oneInchData._buyAmt,stTokenAmount,_isBridged) (contracts/polido/main.sol#273-278)
Reentrancy in ChildPool.arriveShuttle(uint256) (contracts/pools/ChildPool.sol#166-240):
External calls:
- (shuttleNumber,amount,shuttleProcessingStatus) = childTunnel.readData() (contracts/pools/ChildPool.sol#180-184)
- stMaticToken.transfer(feeBeneficiary,shuttleFee) (contracts/pools/ChildPool.sol#212)
- fundCollector.withdrawFunds(amount) (contracts/pools/ChildPool.sol#218)
Event emitted after the call(s):
- ShuttleArrived(shuttleNumber,amount,shuttles[_shuttleNumber].status,shuttleFee) (contracts/pools/ChildPool.sol#234-239)
Reentrancy in RootPool.cancelShuttle(bytes) (contracts/pools/RootPool.sol#143-185):
External calls:
- rootTunnel.receiveMessage(_messageReceiveData) (contracts/pools/RootPool.sol#151)
- (shuttleNumber,amount) = rootTunnel.readData() (contracts/pools/RootPool.sol#154)
- withdrawManagerProxy.processExits(address(maticToken)) (contracts/pools/RootPool.sol#159)
- maticToken.approve(address(depositManagerProxy),amount) (contracts/pools/RootPool.sol#166)
- depositManagerProxy.depositERC20ForUser(address(maticToken),childPoolFundCollector,amount) (contracts/pools/RootPool.sol#168-172)
- rootTunnel.sendMessageToChild(abi.encode(shuttleNumber,amount,ShuttleProcessingStatus.CANCELLED)) (contracts/pools/RootPool.sol#174-176)
Event emitted after the call(s):
- ShuttleProcessed(shuttleNumber,amount,0,ShuttleProcessingStatus.CANCELLED) (contracts/pools/RootPool.sol#178-183)
Reentrancy in ChildPool.claim(uint256) (contracts/pools/ChildPool.sol#266-313):
External calls:
- stMaticToken.transfer(beneficiary,stMaticAmount) (contracts/pools/ChildPool.sol#295)
Event emitted after the call(s):
- TokenClaimed(_shuttleNumber,address(stMaticToken),address(beneficiary),stMaticAmount) (contracts/pools/ChildPool.sol#297-302)
Reentrancy in RootPool.crossChainStake(bytes) (contracts/pools/RootPool.sol#90-134):
External calls:
- rootTunnel.receiveMessage(_messageReceiveData) (contracts/pools/RootPool.sol#98)
- (shuttleNumber,amount) = rootTunnel.readData() (contracts/pools/RootPool.sol#101)
- withdrawManagerProxy.processExits(address(maticToken)) (contracts/pools/RootPool.sol#106)
- maticToken.approve(address(polidoAdapter),amount) (contracts/pools/RootPool.sol#113)
- stMaticAmount = polidoAdapter.depositForAndBridge(address(this),amount) (contracts/pools/RootPool.sol#116-119)
- rootTunnel.sendMessageToChild(abi.encode(shuttleNumber,stMaticAmount,ShuttleProcessingStatus.PROCESSED)) (contracts/pools/RootPool.sol#120-126)
Event emitted after the call(s):
- ShuttleProcessed(shuttleNumber,amount,stMaticAmount,ShuttleProcessingStatus.PROCESSED) (contracts/pools/RootPool.sol#128-133)
Reentrancy in PoLidoAdapter.deposit(uint256) (contracts/polido/main.sol#45-53):
External calls:
- stTokenAmount = _stake(_amount) (contracts/polido/main.sol#46)
- returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#93)
- maticToken.safeTransferFrom(msg.sender,address(this),_amount) (contracts/polido/main.sol#288)
- maticToken.safeApprove(address(stMaticProxy),0) (contracts/polido/main.sol#290)
- maticToken.safeApprove(address(stMaticProxy),_amount) (contracts/polido/main.sol#291)
- stTokenAmount = stMaticProxy.submit(_amount) (contracts/polido/main.sol#292)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
- IERC20Upgradeable(address(stMaticProxy)).safeTransfer(msg.sender,stTokenAmount) (contracts/polido/main.sol#47-50)
External calls sending eth:
- stTokenAmount = _stake(_amount) (contracts/polido/main.sol#46)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Event emitted after the call(s):
- Deposited(msg.sender,_amount,stTokenAmount,false) (contracts/polido/main.sol#52)
Reentrancy in PoLidoAdapter.depositFor(address,uint256) (contracts/polido/main.sol#61-73):
External calls:
- stTokenAmount = _stake(_amount) (contracts/polido/main.sol#66)
- returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#93)
- maticToken.safeTransferFrom(msg.sender,address(this),_amount) (contracts/polido/main.sol#288)
- maticToken.safeApprove(address(stMaticProxy),0) (contracts/polido/main.sol#290)
- maticToken.safeApprove(address(stMaticProxy),_amount) (contracts/polido/main.sol#291)
- stTokenAmount = stMaticProxy.submit(_amount) (contracts/polido/main.sol#292)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
- IERC20Upgradeable(address(stMaticProxy)).safeTransfer(_beneficiary,stTokenAmount) (contracts/polido/main.sol#67-70)
External calls sending eth:
- stTokenAmount = _stake(_amount) (contracts/polido/main.sol#66)
- (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Event emitted after the call(s):
- Deposited(_beneficiary,_amount,stTokenAmount,false) (contracts/polido/main.sol#72)
Reentrancy in ChildPool.enrouteShuttle(uint256) (contracts/pools/ChildPool.sol#124-148):
External calls:
- maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
- childTunnel.sendMessageToRoot(abi.encode(enroutedShuttle,amount)) (contracts/pools/ChildPool.sol#143)
External calls sending eth:
- maticToken.withdraw{value: amount}(amount) (contracts/pools/ChildPool.sol#142)
Event emitted after the call(s):
- ShuttleCreated(currentShuttle) (contracts/pools/ChildPool.sol#87)
- createNewShuttle() (contracts/pools/ChildPool.sol#145)
- ShuttleEnrouted(enroutedShuttle,amount) (contracts/pools/ChildPool.sol#147)
Reentrancy in RootPool.startExitWithBurntTokens(uint256,bytes) (contracts/pools/RootPool.sol#72-81):
External calls:
- erc20PredicateBurnOnly.startExitWithBurntTokens(_burnTokenData) (contracts/pools/RootPool.sol#78)
Event emitted after the call(s):
- ShuttleProcessingInitiated(_shuttleNumber) (contracts/pools/RootPool.sol#80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
ExitPayloadReader.copy(uint256,uint256,uint256) (contracts/lib/ExitPayloadReader.sol#31-65) uses assembly
- INLINE ASM (contracts/lib/ExitPayloadReader.sol#40-42)
- INLINE ASM (contracts/lib/ExitPayloadReader.sol#50-54)
ExitPayloadReader.getReceipt(ExitPayloadReader.ExitPayload) (contracts/lib/ExitPayloadReader.sol#87-111) uses assembly
- INLINE ASM (contracts/lib/ExitPayloadReader.sol#100-103)
Merkle.checkMembership(bytes32,uint256,bytes32,bytes) (contracts/lib/Merkle.sol#5-33) uses assembly
- INLINE ASM (contracts/lib/Merkle.sol#20-22)
RLPReader.toRlpItem(bytes) (contracts/lib/RLPReader.sol#52-59) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#54-56)
RLPReader.isList(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#108-119) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#113-115)
RLPReader.rlpBytesKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#125-133) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#129-131)
RLPReader.payloadKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#146-153) uses assembly


..racts-private
-zsh ...
..uttle-develop
- INLINE ASM (contracts/lib/RLPReader.sol#113-115)
RLPReader.rlpBytesKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#125-133) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#129-131)
RLPReader.payloadKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#146-153) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#149-151)
RLPReader.toRlpBytes(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#158-169) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#163-165)
RLPReader.toBoolean(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#172-181) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#176-178)
RLPReader.toUint(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#190-208) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#198-205)
RLPReader.toUintStrict(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#211-222) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#217-219)
RLPReader.toBytes(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#224-238) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#232-234)
RLPReader._itemLength(uint256) (contracts/lib/RLPReader.sol#260-298) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#263-265)
- INLINE ASM (contracts/lib/RLPReader.sol#270-276)
- INLINE ASM (contracts/lib/RLPReader.sol#280-286)
RLPReader._payloadOffset(uint256) (contracts/lib/RLPReader.sol#293-305) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#295-297)
RLPReader.copy(uint256,uint256,uint256) (contracts/lib/RLPReader.sol#312-339) uses assembly
- INLINE ASM (contracts/lib/RLPReader.sol#321-323)
- INLINE ASM (contracts/lib/RLPReader.sol#334-338)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

FxBaseRootTunnel._validateAndExtractMessage(bytes) (contracts/tunnel/FxBaseRootTunnel.sol#74-127) compares to a boolean constant:
-require(bool,string)(processedExits[exitHash] == false,FxRootTunnel: EXIT_ALREADY_PROCESSED) (contracts/tunnel/FxBaseRootTunnel.sol#91)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

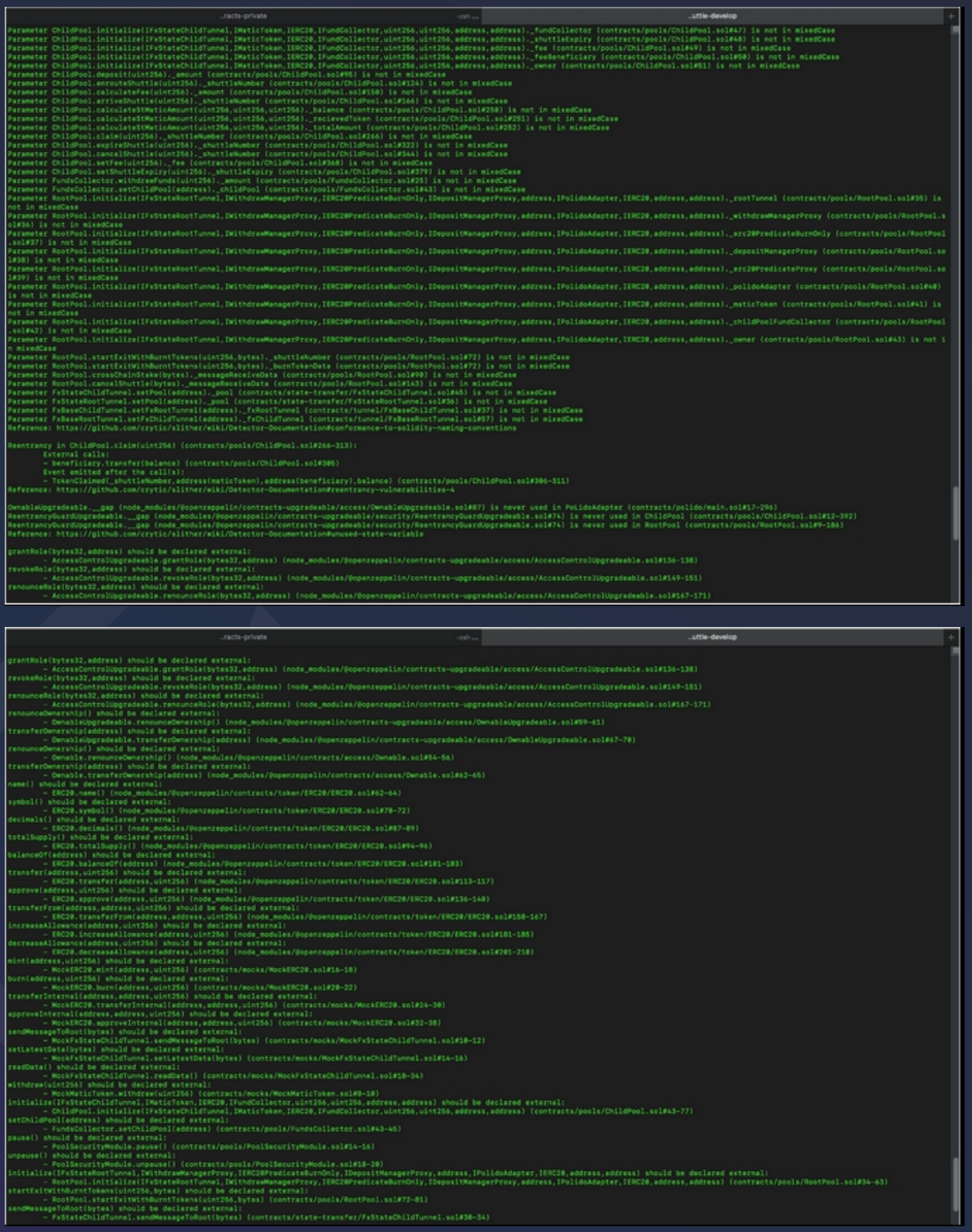
Different versions of Solidity is used:
- Version used: ['^0.8.0', '^0.8.1', '^0.8.3']
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/interfaces/IERC20Upgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
- ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4)
- ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeMath.sol#4)
- ^0.8.0 (contracts/lib/ExitPayloadReader.sol#1)
- ^0.8.0 (contracts/lib/Merkle.sol#2)
- ^0.8.0 (contracts/lib/MerklePatriciaProof.sol#2)
- ^0.8.0 (contracts/lib/RLPReader.sol#5)
- ^0.8.3 (contracts/lib/TokenInterface.sol#3)
- ^0.8.0 (contracts/mocks/MockERC20.sol#1)
- ^0.8.3 (contracts/mocks/MockFxStateChildTunnel.sol#2)
- ^0.8.3 (contracts/mocks/MockMaticToken.sol#2)
- ^0.8.3 (contracts/polido/basic.sol#2)
- ^0.8.3 (contracts/polido/helpers.sol#2)
- ^0.8.3 (contracts/polido/interface.sol#2)
- ^0.8.3 (contracts/polido/main.sol#2)
- ^0.8.3 (contracts/polido/math.sol#2)
- ^0.8.3 (contracts/pools/ChildPool.sol#1)
- ^0.8.3 (contracts/pools/FundsCollector.sol#1)
- ^0.8.3 (contracts/pools/IChildPool.sol#1)
- ^0.8.3 (contracts/pools/IRootPool.sol#1)
- ^0.8.3 (contracts/pools/PoolSecurityModule.sol#1)
- ^0.8.3 (contracts/pools/RootPool.sol#1)
- ^0.8.0 (contracts/state-transfer/FxStateChildTunnel.sol#2)
- ^0.8.0 (contracts/state-transfer/FxStateRootTunnel.sol#2)
- ^0.8.0 (contracts/tunnel/FxBaseChildTunnel.sol#2)
- ^0.8.0 (contracts/tunnel/FxBaseRootTunnel.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

DSMath.add(uint256,uint256) (contracts/polido/math.sol#10-12) is never used and should be removed
DSMath.div(uint256,uint256) (contracts/polido/math.sol#27-29) is never used and should be removed
DSMath.rdiv(uint256,uint256) (contracts/polido/math.sol#39-41) is never used and should be removed
DSMath.rmul(uint256,uint256) (contracts/polido/math.sol#43-45) is never used and should be removed
DSMath.toInt(uint256) (contracts/polido/math.sol#47-50) is never used and should be removed
DSMath.toRad(uint256) (contracts/polido/math.sol#52-54) is never used and should be removed
DSMath.wdiv(uint256,uint256) (contracts/polido/math.sol#35-37) is never used and should be removed
ExitPayloadReader.getBranchMaskAsUint(ExitPayloadReader.ExitPayload) (contracts/lib/ExitPayloadReader.sol#121-123) is never used and should be removed
ExitPayloadReader.toRlpBytes(ExitPayloadReader.Log) (contracts/lib/ExitPayloadReader.sol#152-154) is never used and should be removed
FxBaseChildTunnel._processMessageFromRoot(uint256,address,bytes) (contracts/tunnel/FxBaseChildTunnel.sol#73-77) is never used and should be removed
FxBaseRootTunnel._processMessageFromChild(bytes) (contracts/tunnel/FxBaseRootTunnel.sol#178) is never used and should be removed
RLPReader.hasNext(RLPReader.Iterator) (contracts/lib/RLPReader.sol#44-47) is never used and should be removed
RLPReader.iterator(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#66-71) is never used and should be removed
RLPReader.next(RLPReader.Iterator) (contracts/lib/RLPReader.sol#29-37) is never used and should be removed
RLPReader.payloadKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#146-153) is never used and should be removed
RLPReader.payloadLen(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#83-85) is never used and should be removed
RLPReader.payloadLocation(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#135-140) is never used and should be removed
RLPReader.rlpBytesKeccak256(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#125-133) is never used and should be removed
RLPReader.rlpLen(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#76-78) is never used and should be removed
RLPReader.toBoolean(RLPReader.RLPItem) (contracts/lib/RLPReader.sol#172-181) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/interfaces/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions

```
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/interfaces/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeMath.sol#4) allows old versions
Pragma version^0.8.0 (contracts/lib/ExitPayloadReader.sol#1) allows old versions
Pragma version^0.8.0 (contracts/lib/Merkle.sol#2) allows old versions
Pragma version^0.8.0 (contracts/lib/MerklePatriciaProof.sol#2) allows old versions
Pragma version^0.8.0 (contracts/lib/RLPReader.sol#5) allows old versions
Pragma version^0.8.3 (contracts/lib/TokenInterface.sol#3) allows old versions
Pragma version^0.8.0 (contracts/mocks/MockERC20.sol#1) allows old versions
Pragma version^0.8.3 (contracts/mocks/MockFxStateChildTunnel.sol#2) allows old versions
Pragma version^0.8.3 (contracts/mocks/MockMaticToken.sol#2) allows old versions
Pragma version^0.8.3 (contracts/polido/basic.sol#2) allows old versions
Pragma version^0.8.3 (contracts/polido/helpers.sol#2) allows old versions
Pragma version^0.8.3 (contracts/polido/interface.sol#2) allows old versions
Pragma version^0.8.3 (contracts/polido/main.sol#2) allows old versions
Pragma version^0.8.3 (contracts/polido/math.sol#2) allows old versions
Pragma version^0.8.3 (contracts/pools/ChildPool.sol#1) allows old versions
Pragma version^0.8.3 (contracts/pools/FundsCollector.sol#1) allows old versions
Pragma version^0.8.3 (contracts/pools/IChildPool.sol#1) allows old versions
Pragma version^0.8.3 (contracts/pools/IRootPool.sol#1) allows old versions
Pragma version^0.8.3 (contracts/pools/PoolSecurityModule.sol#1) allows old versions
Pragma version^0.8.3 (contracts/pools/RootPool.sol#1) allows old versions
Pragma version^0.8.0 (contracts/state-transfer/FxStateChildTunnel.sol#2) allows old versions
Pragma version^0.8.0 (contracts/state-transfer/FxStateRootTunnel.sol#2) allows old versions
Pragma version^0.8.0 (contracts/tunnel/FxBaseChildTunnel.sol#2) allows old versions
Pragma version^0.8.0 (contracts/tunnel/FxBaseRootTunnel.sol#2) allows old versions
solc-0.8.3 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Low level call in PoLidoAdapter.oneInchSwap(OneInchData,uint256) (contracts/polido/main.sol#182-212):
        - (success) = oneInchAddr.call{value: ethAmt}(oneInchData.callData) (contracts/polido/main.sol#202-204)
Low level call in FundsCollector.withdrawFunds(uint256) (contracts/pools/FundsCollector.sol#25-36):
        - (sent) = childPool.call{value: _amount}() (contracts/pools/FundsCollector.sol#33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
MockFxStateChildTunnel (contracts/mocks/MockFxStateChildTunnel.sol#6-35) should inherit from IFxStateChildTunnel (contracts/pools/IChildPool.sol#46-56)
MockMaticToken (contracts/mocks/MockMaticToken.sol#4-12) should inherit from IMaticToken (contracts/pools/IChildPool.sol#58-60)
PoLidoAdapter (contracts/polido/main.sol#17-296) should inherit from IPoLidoAdapter (contracts/pools/IRootPool.sol#47-51)
FundsCollector (contracts/pools/FundsCollector.sol#12-50) should inherit from IFundCollector (contracts/pools/IChildPool.sol#62-64)
FxStateChildTunnel (contracts/state-transfer/FxStateChildTunnel.sol#12-48) should inherit from IFxStateChildTunnel (contracts/pools/IChildPool.sol#46-56)
FxStateRootTunnel (contracts/state-transfer/FxStateRootTunnel.sol#11-39) should inherit from IFxStateRootTunnel (contracts/pools/IRootPool.sol#23-29)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-inheritance

Function AccessControlUpgradeable.__AccessControl_init() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#235) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol#102) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Parameter MockFxStateChildTunnel.sendMessageToRoot(bytes)._data (contracts/mocks/MockFxStateChildTunnel.sol#10) is not in mixedCase
Parameter MockFxStateChildTunnel.setLatestData(bytes)._latestData (contracts/mocks/MockFxStateChildTunnel.sol#14) is not in mixedCase
Parameter MockMaticToken.withdraw(uint256)._amount (contracts/mocks/MockMaticToken.sol#8) is not in mixedCase
Parameter Basic.convert18ToDec(uint256,uint256)._dec (contracts/polido/basic.sol#13) is not in mixedCase
Parameter Basic.convert18ToDec(uint256,uint256)._amt (contracts/polido/basic.sol#13) is not in mixedCase
Parameter Basic.convertTo18(uint256,uint256)._dec (contracts/polido/basic.sol#21) is not in mixedCase
Parameter Basic.convertTo18(uint256,uint256)._amt (contracts/polido/basic.sol#21) is not in mixedCase
Constant Basic.ethAddr (contracts/polido/basic.sol#10-11) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Helpers.maticToken (contracts/polido/helpers.sol#13-14) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Helpers.stMaticProxy (contracts/polido/helpers.sol#15-16) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Helpers.rootChainManagerProxy (contracts/polido/helpers.sol#18-19) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Helpers.mintableERC20Proxy (contracts/polido/helpers.sol#21-22) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Helpers.oneInchAddr (contracts/polido/helpers.sol#27-28) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter PoLidoAdapter.initialize(uint256)._feePercentage (contracts/polido/main.sol#38) is not in mixedCase
Parameter PoLidoAdapter.changeFeePercentage(uint256)._newFeePercentage (contracts/polido/main.sol#35) is not in mixedCase
Parameter PoLidoAdapter.deposit(uint256)._amount (contracts/polido/main.sol#45) is not in mixedCase
Parameter PoLidoAdapter.depositFor(address,uint256)._beneficiary (contracts/polido/main.sol#61) is not in mixedCase
Parameter PoLidoAdapter.depositFor(address,uint256)._amount (contracts/polido/main.sol#61) is not in mixedCase
Parameter PoLidoAdapter.swapAndStake(address,uint256,uint256,bytes)._sellAddr (contracts/polido/main.sol#85) is not in mixedCase
Parameter PoLidoAdapter.swapAndStake(address,uint256,uint256,bytes)._sellAmt (contracts/polido/main.sol#86) is not in mixedCase
Parameter PoLidoAdapter.swapAndStake(address,uint256,uint256,bytes)._unitAmt (contracts/polido/main.sol#87) is not in mixedCase
Parameter PoLidoAdapter.swapAndStake(address,uint256,uint256,bytes)._callData (contracts/polido/main.sol#88) is not in mixedCase
Parameter PoLidoAdapter.depositForAndBridge(address,uint256)._beneficiary (contracts/polido/main.sol#111) is not in mixedCase
Parameter PoLidoAdapter.depositForAndBridge(address,uint256)._amount (contracts/polido/main.sol#111) is not in mixedCase
Parameter PoLidoAdapter.swapStakeAndBridge(address,uint256,uint256,bytes,address)._sellAddr (contracts/polido/main.sol#132) is not in mixedCase
Parameter PoLidoAdapter.swapStakeAndBridge(address,uint256,uint256,bytes,address)._sellAmt (contracts/polido/main.sol#133) is not in mixedCase
Parameter PoLidoAdapter.swapStakeAndBridge(address,uint256,uint256,bytes,address)._unitAmt (contracts/polido/main.sol#134) is not in mixedCase
Parameter PoLidoAdapter.swapStakeAndBridge(address,uint256,uint256,bytes,address)._callData (contracts/polido/main.sol#135) is not in mixedCase
Parameter PoLidoAdapter.swapStakeAndBridge(address,uint256,uint256,bytes,address)._beneficiary (contracts/polido/main.sol#136) is not in mixedCase
Parameter PoLidoAdapter.calculateFee(uint256)._amt (contracts/polido/main.sol#173) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._childTunnel (contracts/pools/ChildPool.sol#44) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._maticToken (contracts/pools/ChildPool.sol#45) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._stMaticToken (contracts/pools/ChildPool.sol#46) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._fundCollector (contracts/pools/ChildPool.sol#47) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._shuttleExpiry (contracts/pools/ChildPool.sol#48) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._fee (contracts/pools/ChildPool.sol#49) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._feeBeneficiary (contracts/pools/ChildPool.sol#50) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._owner (contracts/pools/ChildPool.sol#51) is not in mixedCase
Parameter ChildPool.deposit(uint256)._amount (contracts/pools/ChildPool.sol#95) is not in mixedCase
```


..racts-private
..ttle-develop
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._fundCollector (contracts/pools/ChildPool.sol#47) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._shuttleExpiry (contracts/pools/ChildPool.sol#48) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._fee (contracts/pools/ChildPool.sol#49) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._feeBeneficiary (contracts/pools/ChildPool.sol#50) is not in mixedCase
Parameter ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address)._owner (contracts/pools/ChildPool.sol#51) is not in mixedCase
Parameter ChildPool.deposit(uint256)._amount (contracts/pools/ChildPool.sol#95) is not in mixedCase
Parameter ChildPool.enrouteShuttle(uint256)._shuttleNumber (contracts/pools/ChildPool.sol#124) is not in mixedCase
Parameter ChildPool.calculateFee(uint256)._amount (contracts/pools/ChildPool.sol#150) is not in mixedCase
Parameter ChildPool.arriveShuttle(uint256)._shuttleNumber (contracts/pools/ChildPool.sol#166) is not in mixedCase
Parameter ChildPool.calculateStMaticAmount(uint256,uint256,uint256)._balance (contracts/pools/ChildPool.sol#250) is not in mixedCase
Parameter ChildPool.calculateStMaticAmount(uint256,uint256,uint256)._recievedToken (contracts/pools/ChildPool.sol#251) is not in mixedCase
Parameter ChildPool.calculateStMaticAmount(uint256,uint256,uint256)._totalAmount (contracts/pools/ChildPool.sol#252) is not in mixedCase
Parameter ChildPool.claim(uint256)._shuttleNumber (contracts/pools/ChildPool.sol#266) is not in mixedCase
Parameter ChildPool.expireShuttle(uint256)._shuttleNumber (contracts/pools/ChildPool.sol#322) is not in mixedCase
Parameter ChildPool.cancelShuttle(uint256)._shuttleNumber (contracts/pools/ChildPool.sol#344) is not in mixedCase
Parameter ChildPool.setFee(uint256)._fee (contracts/pools/ChildPool.sol#368) is not in mixedCase
Parameter ChildPool.setShuttleExpiry(uint256)._shuttleExpiry (contracts/pools/ChildPool.sol#379) is not in mixedCase
Parameter FundsCollector.withdrawFunds(uint256)._amount (contracts/pools/FundsCollector.sol#25) is not in mixedCase
Parameter FundsCollector.setChildPool(address)._childPool (contracts/pools/FundsCollector.sol#43) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._rootTunnel (contracts/pools/RootPool.sol#35) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._withdrawManagerProxy (contracts/pools/RootPool.sol#36) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._erc20PredicateBurnOnly (contracts/pools/RootPool.sol#37) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._depositManagerProxy (contracts/pools/RootPool.sol#38) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._erc20PredicateProxy (contracts/pools/RootPool.sol#39) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._polidoAdapter (contracts/pools/RootPool.sol#40) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._maticToken (contracts/pools/RootPool.sol#41) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._childPoolFundCollector (contracts/pools/RootPool.sol#42) is not in mixedCase
Parameter RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address)._owner (contracts/pools/RootPool.sol#43) is not in mixedCase
Parameter RootPool.startExitWithBurntTokens(uint256,bytes)._shuttleNumber (contracts/pools/RootPool.sol#72) is not in mixedCase
Parameter RootPool.startExitWithBurntTokens(uint256,bytes)._burnTokenData (contracts/pools/RootPool.sol#72) is not in mixedCase
Parameter RootPool.crossChainStake(bytes)._messageReceiveData (contracts/pools/RootPool.sol#90) is not in mixedCase
Parameter RootPool.cancelShuttle(bytes)._messageReceiveData (contracts/pools/RootPool.sol#143) is not in mixedCase
Parameter FxStateChildTunnel.setPool(address)._pool (contracts/state-transfer/FxStateChildTunnel.sol#45) is not in mixedCase
Parameter FxStateRootTunnel.setPool(address)._pool (contracts/state-transfer/FxStateRootTunnel.sol#36) is not in mixedCase
Parameter FxBaseChildTunnel.setFxRootTunnel(address)._fxRootTunnel (contracts/tunnel/FxBaseChildTunnel.sol#37) is not in mixedCase
Parameter FxBaseRootTunnel.setFxChildTunnel(address)._fxChildTunnel (contracts/tunnel/FxBaseRootTunnel.sol#57) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Reentrancy in ChildPool.claim(uint256) (contracts/pools/ChildPool.sol#266-313):
External calls:
- beneficiary.transfer(balance) (contracts/pools/ChildPool.sol#305)
Event emitted after the call(s):
- TokenClaimed(_shuttleNumber,address(maticToken),address(beneficiary),balance) (contracts/pools/ChildPool.sol#306-311)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#87) is never used in PoLidoAdapter (contracts/polido/main.sol#17-296)
ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is never used in ChildPool (contracts/pools/ChildPool.sol#12-392)
ReentrancyGuardUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is never used in RootPool (contracts/pools/RootPool.sol#9-186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
grantRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#136-138)
revokeRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#149-151)
renounceRole(bytes32,address) should be declared external:
- AccessControlUpgradeable.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts-upgradeable/access/AccessControlUpgradeable.sol#167-171)
renounceOwnership() should be declared external:
- OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#59-61)
transferOwnership(address) should be declared external:
- OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#67-70)
renounceOwnership() should be declared external:
- Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-56)
transferOwnership(address) should be declared external:
- Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#62-65)
name() should be declared external:
- ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
- ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
- ERC20.decimals() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
- ERC20.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
- ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
- ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
- ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#136-140)
transferFrom(address,address,uint256) should be declared external:
- ERC20.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#158-167)
increaseAllowance(address,uint256) should be declared external:
- ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
- ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#201-218)
mint(address,uint256) should be declared external:
- MockERC20.mint(address,uint256) (contracts/mocks/MockERC20.sol#16-18)
burn(address,uint256) should be declared external:
- MockERC20.burn(address,uint256) (contracts/mocks/MockERC20.sol#20-22)
transferInternal(address,address,uint256) should be declared external:
- MockERC20.transferInternal(address,address,uint256) (contracts/mocks/MockERC20.sol#24-30)
approveInternal(address,address,uint256) should be declared external:
- MockERC20.approveInternal(address,address,uint256) (contracts/mocks/MockERC20.sol#32-38)
sendMessageToRoot(bytes) should be declared external:
- MockFxStateChildTunnel.sendMessageToRoot(bytes) (contracts/mocks/MockFxStateChildTunnel.sol#10-12)
setLatestData(bytes) should be declared external:
- MockFxStateChildTunnel.setLatestData(bytes) (contracts/mocks/MockFxStateChildTunnel.sol#14-16)
readData() should be declared external:
- MockFxStateChildTunnel.readData() (contracts/mocks/MockFxStateChildTunnel.sol#18-34)
withdraw(uint256) should be declared external:
- MockMaticToken.withdraw(uint256) (contracts/mocks/MockMaticToken.sol#8-10)
initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address) should be declared external:
- ChildPool.initialize(IFxStateChildTunnel,IMaticToken,IERC20,IFundCollector,uint256,uint256,address,address) (contracts/pools/ChildPool.sol#43-77)
setChildPool(address) should be declared external:
- FundsCollector.setChildPool(address) (contracts/pools/FundsCollector.sol#43-45)
pause() should be declared external:
- PoolSecurityModule.pause() (contracts/pools/PoolSecurityModule.sol#14-16)
unpause() should be declared external:
- PoolSecurityModule.unpause() (contracts/pools/PoolSecurityModule.sol#18-20)
initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address) should be declared external:
- RootPool.initialize(IFxStateRootTunnel,IWithdrawManagerProxy,IERC20PredicateBurnOnly,IDepositManagerProxy,address,IPolidoAdapter,IERC20,address,address) (contracts/pools/RootPool.sol#34-63)
startExitWithBurntTokens(uint256,bytes) should be declared external:
- RootPool.startExitWithBurntTokens(uint256,bytes) (contracts/pools/RootPool.sol#72-81)
sendMessageToRoot(bytes) should be declared external:
- FxStateChildTunnel.sendMessageToRoot(bytes) (contracts/state-transfer/FxStateChildTunnel.sol#30-34)

Closing Summary

In this report, we have considered the security of the Stakeall smart contracts. We performed our audit according to the procedure described above.

Some issues of Medium, Low and informational severity were found, some suggestions and best practices are also provided in order to improve the code quality and security posture.

In the End,Stakeall Team Resolved all issues

Disclaimer

QuillAudits smart contract audit is not a security warranty, investment advice, or an endorsement of the Stakeall Platform. This audit does not provide a security or correctness guarantee of the audited smart contracts.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the Stakeall Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

Follow Our Journey

Audit Report
May, 2022

For

STAKEALL

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com