Aderyn Analysis Report

This report was generated by Aderyn, a static analysis tool built by Cyfrin, a blockchain security company. This report is not a substitute for manual audit or security review. It should not be relied upon for any purpose other than to assist in the identification of potential security vulnerabilities.

Table of Contents

Summary

Files Summary

Key	Value
.sol Files	18
Total nSLOC	847

Files Details

Filepath	nSLOC
src/abstract/AStaticTokenData.sol	14
src/abstract/AStaticUSDCData.sol	14
src/abstract/AStaticWethData.sol	13
src/dao/VaultGuardianGovernor.sol	26
src/dao/VaultGuardianToken.sol	17

Filepath	nSLOC
src/interfaces/IVaultData.sol	8
src/interfaces/IVaultGuardians.sol	2
src/interfaces/IVaultShares.sol	21
src/interfaces/InvestableUniverseAdapter.sol	4
src/protocol/VaultGuardians.sol	34
src/protocol/VaultGuardiansBase.sol	176
src/protocol/VaultShares.sol	151
src/protocol/investableUniverseAdapters/AaveAdapter.sol	31
src/protocol/investableUniverseAdapters/UniswapAdapter.sol	75
src/vendor/DataTypes.sol	204
src/vendor/IPool.sol	7
src/vendor/IUniswapV2Factory.sol	12
src/vendor/IUniswapV2Router01.sol	38
Total	**847**

Issue Summary

Category	No. of Issues
High	1
Low	8

High Issues

H-1: Using `block.timestamp` for swap deadline offers no protection

In the PoS model, proposers know well in advance if they will propose one or consecutive blocks ahead of time. In such a scenario, a malicious validator can hold back the transaction and execute it at a more favourable block number.Consider allowing function caller to specify swap deadline input parameter.

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 40

```
        uint256[] memory amounts =
i_uniswapRouter.swapExactTokensForTokens({
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 84

```
        uint256[] memory amounts =
i_uniswapRouter.swapExactTokensForTokens({
```

Low Issues

L-1: Centralization Risk for trusted owners

Contracts have owners with privileged rights to perform admin tasks and need to be trusted to not perform malicious updates or drain funds.

- Found in src/dao/VaultGuardianToken.sol Line: 9

```
contract VaultGuardianToken is ERC20, ERC20Permit, ERC20Votes, Ownable {
```

- Found in src/dao/VaultGuardianToken.sol Line: 21

```
    function mint(address to, uint256 amount) external onlyOwner {
```

- Found in src/protocol/VaultGuardians.sol Line: 40

```
contract VaultGuardians is Ownable, VaultGuardiansBase {
```

- Found in src/protocol/VaultGuardians.sol Line: 71

```
    function updateGuardianStakePrice(uint256 newStakePrice) external
onlyOwner {
```

- Found in src/protocol/VaultGuardians.sol Line: 82

```
    function updateGuardianAndDaoCut(uint256 newCut) external onlyOwner {
```

L-2: Unsafe ERC20 Operations should not be used

ERC20 functions may not behave as expected. For example: return values are not always meaningful. It is recommended to use OpenZeppelin's SafeERC20 library.

- Found in src/protocol/VaultGuardiansBase.sol Line: 257

```

```

```
        bool succ = token.approve(address(tokenVault),
s_guardianStakePrice);
```

- Found in src/protocol/investableUniverseAdapters/AaveAdapter.sol Line: 20

```
        bool succ = asset.approve(address(i_aavePool), amount);
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 36

```
        bool succ = token.approve(address(i_uniswapRouter),
amountOfTokenToSwap);
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 48

```
        succ = counterPartyToken.approve(address(i_uniswapRouter),
amounts[1]);
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 52

```
        succ = token.approve(address(i_uniswapRouter), amountOfTokenToSwap +
amounts[0]);
```

L-3: Missing checks for `address(0)` when assigning values to address state variables

Check for `address(0)` when assigning values to address state variables.

- Found in src/protocol/VaultGuardiansBase.sol Line: 253

```
        s_guardians[msg.sender][token] = IVaultShares(address(tokenVault));
```

L-4: `public` functions not used internally could be marked `external`

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

- Found in src/dao/VaultGuardianGovernor.sol Line: 17

```
    function votingDelay() public pure override returns (uint256) {
```

- Found in src/dao/VaultGuardianGovernor.sol Line: 21

```
    function votingPeriod() public pure override returns (uint256) {
```

- Found in src/dao/VaultGuardianGovernor.sol Line: 27

```
    function quorum(uint256 blockNumber)
```

- Found in src/dao/VaultGuardianToken.sol Line: 17

```
    function nonces(address ownerOfNonce) public view override(ERC20Permit,
Nonces) returns (uint256) {
```

- Found in src/protocol/VaultShares.sol Line: 108

```
    function setNotActive() public onlyVaultGuardians isActive {
```

- Found in src/protocol/VaultShares.sol Line: 129

```
    function deposit(uint256 assets, address receiver)
```

- Found in src/protocol/VaultShares.sol Line: 166

```
    function rebalanceFunds() public isActive divestThenInvest nonReentrant
{}
```

- Found in src/protocol/VaultShares.sol Line: 174

```
    function withdraw(uint256 assets, address receiver, address owner)
```

- Found in src/protocol/VaultShares.sol Line: 191

```
    function redeem(uint256 shares, address receiver, address owner)
```

L-5: Event is missing `indexed` fields

Index event fields make the field more quickly accessible to off-chain tools that parse events. However, note that each index field costs extra gas during emission, so it's not necessarily best to index the maximum allowed per event (three fields). Each event should use three indexed fields if there are three or more fields, and gas usage is not particularly of concern for the events in question. If there are fewer than three fields, all of the fields should be indexed.

- Found in src/protocol/VaultGuardians.sol Line: 48

```
    event VaultGuardians__UpdatedStakePrice(uint256 oldStakePrice, uint256
newStakePrice);
```

- Found in src/protocol/VaultGuardians.sol Line: 49

```
    event VaultGuardians__UpdatedFee(uint256 oldFee, uint256 newFee);
```

- Found in src/protocol/VaultGuardians.sol Line: 50

```
    event VaultGuardians__SweptTokens(address asset);
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 78

```
    event GuardianAdded(address guardianAddress, IERC20 token);
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 79

```
    event GaurdianRemoved(address guardianAddress, IERC20 token);
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 80

```
    event InvestedInGuardian(address guardianAddress, IERC20 token, uint256
amount);
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 81

```
    event DinvestedFromGuardian(address guardianAddress, IERC20 token,
uint256 amount);
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 82

```
    event GuardianUpdatedHoldingAllocation(address guardianAddress, IERC20
token);
```

- Found in src/protocol/VaultShares.sol Line: 35

```
    event UpdatedAllocation(AllocationData allocationData);
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 19

```
    event UniswapInvested(uint256 tokenAmount, uint256 wethAmount, uint256
liquidity);
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 20

```
    event UniswapDivested(uint256 tokenAmount, uint256 wethAmount);
```

- Found in src/vendor/IUniswapV2Factory.sol Line: 7

```
    event PairCreated(address indexed token0, address indexed token1,
address pair, uint256);
```

L-6: The `nonReentrant modifier` should occur before all other modifiers

This is a best-practice to protect against reentrancy in other modifiers.

- Found in src/protocol/VaultShares.sol Line: 133

```
        nonReentrant
```

- Found in src/protocol/VaultShares.sol Line: 166

```
    function rebalanceFunds() public isActive divestThenInvest nonReentrant
{}
```

- Found in src/protocol/VaultShares.sol Line: 178

```
        nonReentrant
```

- Found in src/protocol/VaultShares.sol Line: 195

```
        nonReentrant
```

L-7: PUSH0 is not supported by all chains

Solc compiler version 0.8.20 switches the default target EVM version to Shanghai, which means that the generated bytecode will include PUSH0 opcodes. Be sure to select the appropriate EVM version in case you intend to deploy on a chain other than mainnet like L2 chains that may not support PUSH0, otherwise deployment of your contracts will fail.

- Found in src/abstract/AStaticTokenData.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/abstract/AStaticUSDCData.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/abstract/AStaticWethData.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/dao/VaultGuardianGovernor.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/dao/VaultGuardianToken.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/interfaces/IVaultData.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/interfaces/IVaultGuardians.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/interfaces/IVaultShares.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/interfaces/InvestableUniverseAdapter.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/protocol/VaultGuardians.sol Line: 28

```
pragma solidity 0.8.20;
```

- Found in src/protocol/VaultGuardiansBase.sol Line: 28

```
pragma solidity 0.8.20;
```

- Found in src/protocol/VaultShares.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/protocol/investableUniverseAdapters/AaveAdapter.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/protocol/investableUniverseAdapters/UniswapAdapter.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/vendor/DataTypes.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/vendor/IPool.sol Line: 2

```
pragma solidity 0.8.20;
```

- Found in src/vendor/IUniswapV2Factory.sol Line: 3

```
pragma solidity 0.8.20;
```

- Found in src/vendor/IUniswapV2Router01.sol Line: 2

```
pragma solidity 0.8.20;
```

L-8: Empty Block

Consider removing empty blocks.

- Found in src/protocol/VaultShares.sol Line: 166

```
    function rebalanceFunds() public isActive divestThenInvest nonReentrant
{}
```