FYEO

Theoriq

Repo: https://github.com/Moonsong-Labs/tq-oracle

Security Review Update: December 1, 2025

Reviewer: thomas@gofyeo.com

Tq-Oracle Ongoing

New security issues, 4

After the development team implemented the latest updates, FYEO conducted a review of the modifications. The primary goal of this evaluation was to ensure the continued robustness of the program's security features, safeguarding user funds and maintaining the overall integrity of the program.

General Updates:

The StrETHAdapter collects strETH staking positions for the oracle's TVL calculations. It fetches all subvault addresses from the vault contract, aggregates balances by asset across all subvaults, and returns the totals for the oracle pipeline.

The adapter follows the general patterns established by existing adapters but lacks several defensive measures that should be added before production use. The highest priority items are:

1. Adding error handling around ABI decode operations
2. Adding RPC connection validation
3. Implementing network-specific address configuration

These changes would bring the adapter in line with the defensive coding practices already present in the StakeWiseAdapter.

HARDCODED CONTRACT ADDRESSES IN `CONSTANTS.PY`

Finding ID: FY-TQO-01
Severity: **Medium**
Status: **Remediated**

Description

The strETH-related contract addresses are hardcoded in constants.py without network-specific configuration. Unlike StakeWise which has per-network address mappings, strETH assumes mainnet-only deployment. This could cause issues if strETH is deployed on other networks or if addresses need to change.

Proof of Issue

File name: src/tq_oracle/constants.py
Line number: 23-25

```
STRETH = "0x277C6A642564A91ff78b008022D65683cEE5CCC5"
MULTICALL_ADDRESS = "0xcA11bde05977b3631167028862bE2a173976CA11"
CORE_VAULTS_COLLECTOR = "0x551233202dcC8761123c0489c3D59ef602f6BEC6"
```

Severity and Impact Summary

Using hardcoded addresses without network validation could lead to calls against wrong contracts if the oracle is misconfigured for a different network, potentially returning garbage data or failing silently.

Recommendation

Implement network-specific address configuration similar to StakeWise and validate network in the adapter:

Missing Error Handling for Raw ABI Decode Operations in `asset_adapters/streth.py`

Finding ID: FY-TQO-02
Severity: **Medium**
Status: **Remediated**

Description

The _fetch_assets method uses raw eth_abi.decode() to decode bytes returned from the multicall without any error handling. This is unique among the asset adapters — both IdleBalancesAdapter and StakeWiseAdapter rely on web3.py's built-in contract interface which handles ABI decoding automatically. The StrETHAdapter requires manual decoding because it uses multicall's aggregate() function which returns raw bytes.

The other adapters use web3.py's contract interface which handles ABI decoding internally:

Proof of Issue

File name: src/tq_oracle/adapters/asset_adapters/streth.py
Line number: 116-118

```
balances = list(
    decode(["(address,int256,string,address)[]"], call_result)[0]
)
```

Severity and Impact Summary

A malformed response from the CoreVaultsCollector contract (due to upgrade, bug, or malicious RPC) can cause an unhandled exception, halting the entire oracle pipeline and preventing price updates. This risk is elevated because StrETHAdapter is the only adapter performing raw ABI decoding without protection.

Recommendation

Wrap the decode operation in a try/except block with appropriate logging and error handling: Additionally, consider adding similar error handling around the second decode operation at line 160:

```
try:
    subvaults = [decode(["address"], response)[0] for response in responses]
except Exception as e:
    logger.error("Failed to decode subvaultAt responses: %s", e)
    raise ValueError("Invalid subvault address data from multicall") from e
```

NO RPC CONNECTION VALIDATION

Finding ID: FY-TQO-03
Severity: **Medium**
Status: **Remediated**

Description

The adapter initializes a Web3 connection without verifying that the connection is actually established. This is inconsistent with the StakeWiseAdapter which validates connectivity during initialization.

Proof of Issue

File name: src/tq_oracle/adapters/asset_adapters/streth.py
Line number: 37

```
self.w3 = Web3(Web3.HTTPProvider(config.vault_rpc_required))
# No connection check
```

Severity and Impact Summary

If the RPC endpoint is unreachable, errors will only surface when the first RPC call is made, potentially after significant initialization work. Early failure detection provides clearer error messages and faster debugging.

Recommendation

Add connection validation in __init__:

MISSING ZERO ADDRESS VALIDATION ON DECODED ASSETS

Finding ID: FY-TQO-04
Severity: **Low**
Status: **Remediated**

Description

Asset addresses decoded from the contract response are converted to checksum addresses but not validated for being non-zero or having expected format before being added to results.

Proof of Issue

File name: src/tq_oracle/adapters/asset_adapters/streth.py
Line number: 119-127

```
for asset, amount, _, _ in balances:
    if amount != 0:
        cumulative_amounts[asset] = (
            cumulative_amounts.get(asset, 0) + amount
        )

result: list[AssetData] = []
for asset, amount in cumulative_amounts.items():
    result.append(AssetData(Web3.to_checksum_address(asset), amount))
```

Severity and Impact Summary

Invalid or zero addresses in the response could cause unexpected behavior downstream. While `to_checksum_address` will raise on truly invalid data, the zero address would pass through.

Recommendation

Add validation for the zero address:

Commit Hash Reference:

For transparency and reference, the security review was conducted on the specific commit hash for the Theoriq repository. The commit hash for the reviewed versions is as follows:

905aeb40cdbd8a823611be21fc8153862e429d7d

Remediations have been submitted with the commit hash:
98da93bc70c356bbc28bf89a2ddcb8296f2a3428.

Conclusion:

In conclusion, the security aspects of the Theoriq program remain robust and unaffected by the recent updates. Users can confidently interact with the protocol, assured that their funds are well-protected. The commitment to security exhibited by the development team is commendable, and we appreciate the ongoing efforts to prioritize the safeguarding of user assets.