Scan Findings:

Report generated by auditbase.com

Issues

Reentrancy Vulnerability in ETH and ERC20 Flush Functions

Description:

The flushETH and flushERC20 functions transfer ETH or ERC20 tokens to an external address which could potentially execute code that re-enters the flush functions, leading to unexpected behavior or draining of funds.

Severity:

critical

Snippet:

```
(bool success, ) = destination.call{ value: balance }("");
tokenContract.safeTransfer(destination, forwarderBalance);
```

Lack of Access Control on Flush Functions

Description:

The flushETH and flushERC20 functions can be called by any external user, not just the owner or a designated administrator, potentially leading to unauthorized or premature flushing of funds.

Severity:

high

Snippet:

```
function flushETH() external {
function flushERC20(address tokenContractAddress) external {
```

Potential for Setting Zero Address as Destination

Description:

The constructor allows setting the destination address to any value, including the zero address. It would be safer to enforce this in the constructor to prevent the contract from being deployed with an invalid state.

Severity:

medium

Snippet:

```
constructor(address _destination) {
  destination = _destination;
}
```