yAudit EVK Periphery Guardians Review

Review Resources:

- Guardians documentation.

Auditors:

- Invader-Tak
- Adriro

Table of Contents

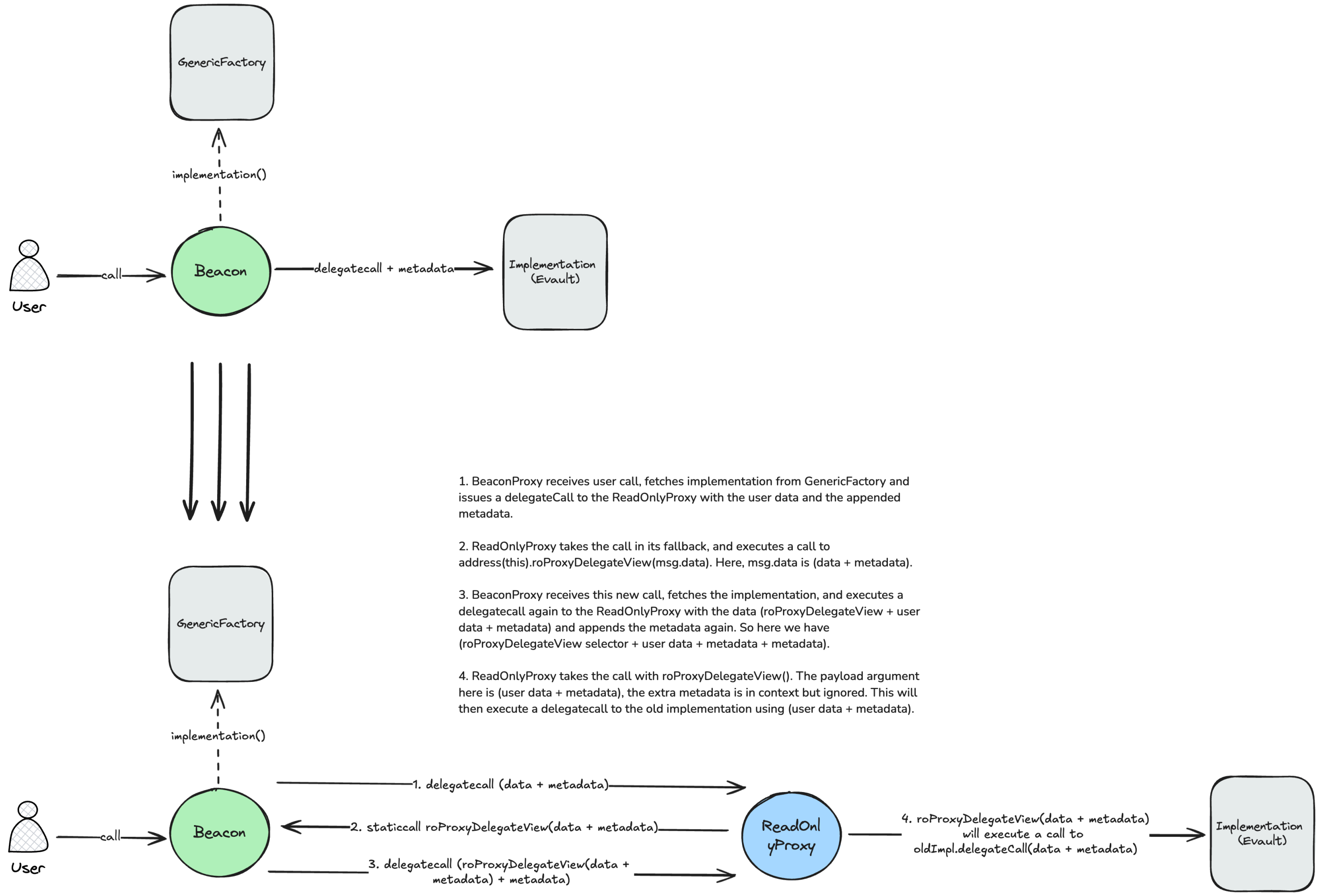
Review Summary

Euler Vault Kit Periphery

The Euler Vault Kit is a system for constructing credit vaults. Credit vaults are ERC-4626 vaults with added borrowing functionality. Unlike typical ERC-4626 vaults, which earn yield by actively investing deposited funds, credit vaults are passive lending pools.

The EVK Periphery codebase has been expanded to include a set of guardian contracts: governors and hooks that have the ability to pause one or more vaults.

The factory governor contract provides an emergency action to pause all upgradeable vaults by redirecting traffic to a read-only proxy. This proxy functions as a middleware that sits between the beacon proxy and the implementation contract, blocking all state-modifying actions by wrapping a `delegatecall` inside a `staticcall`.


GenericFactory
implementation()
User
call
Beacon
delegatecall + metadata
Implementation (Evault)
GenericFactory
implementation()
User
call
Beacon
1. delegatecall (data + metadata)
2. staticcall roProxyDelegateView(data + metadata)
3. delegatecall (roProxyDelegateView(data + metadata) + metadata)
ReadOnl yProxy
4. roProxyDelegateView(data + metadata) will execute a call to oldImpl.delegateCall(data + metadata)
Implementation (Evault)

1. BeaconProxy receives user call, fetches implementation from GenericFactory and issues a delegateCall to the ReadOnlyProxy with the user data and the appended metadata.

2. ReadOnlyProxy takes the call in its fallback, and executes a call to address(this).roProxyDelegateView(msg.data). Here, msg.data is (data + metadata).

3. BeaconProxy receives this new call, fetches the implementation, and executes a delegatecall again to the ReadOnlyProxy with the data (roProxyDelegateView + user data + metadata) and appends the metadata again. So here we have (roProxyDelegateView selector + user data + metadata + metadata).

4. ReadOnlyProxy takes the call with roProxyDelegateView(). The payload argument here is (user data + metadata), the extra metadata is in context but ignored. This will then execute a delegatecall to the old implementation using (user data + metadata).

The contracts of the EVK Periphery repository were reviewed over two days. The code review was performed by two auditors between Aug 8 and Aug 9, 2024. The repository was under active development during the review, but the review was limited to the latest commit at the start of the review. This was commit c8b4af83b37a49d86be379e7427ee325dfdde036 for the EVK Periphery repository.

Scope

The scope of the review consisted of the following contracts at the specific commit:

```
src/Governor
├── FactoryGovernor.sol
├── GovernorGuardian.sol
└── ReadOnlyProxy.sol
src/HookTarget
└── HookTargetGuardian.sol
```

After the findings were presented to the Euler team, fixes were made and included in several PRs.

This review is a code review to identify potential vulnerabilities in the code. The reviewers did not investigate security practices or operational security and assumed that privileged accounts could be trusted. The reviewers did not evaluate the security of the code relative to a standard or specification. The review may not have identified all potential attack vectors or areas of vulnerability.

yAudit and the auditors make no warranties regarding the security of the code and do not warrant that the code is free from defects. yAudit and the auditors do not represent nor imply to third parties that the code has been audited nor that the code is free from defects. By deploying or using the code, Euler and users of the contracts agree to use the code at their own risk.

Code Evaluation Matrix

Category	Mark	Description
Access Control	Good	Proper access control is provided by the AccessControl library.
Mathematics	Good	There are no complex mathematical operations.
Complexity	Good	The contracts are well coded and easy to read.
Libraries	Good	The codebase relies on an updated version of the OpenZeppelin library.
Decentralization	Good	Guardians have limited responsibility. Pauses can be permissionlessly disabled after a certain period.
Code stability	Good	The codebase remained stable during the review.
Documentation	Good	The contracts have high level documentation and NatSpec.
Monitoring	Good	The pausing functionality emits the corresponding events.

Category	Mark	Description
Testing and verification	Good	Code under scope is tested with a good coverage.

Findings Explanation

Findings are broken down into sections by their respective impact:

- Critical, High, Medium, Low impact
  - These are findings that range from attacks that may cause loss of funds, impact control/ownership of the contracts, or cause any unintended consequences/actions that are outside the scope of the requirements.
- Gas savings
  - Findings that can improve the gas efficiency of the contracts.
- Informational
  - Findings including recommendations and best practices.

Critical Findings

None.

High Findings

None.

Medium Findings

None.

Low Findings

1. Low - Hook config can be changed through the EVC

The GovernorGuardian.sol contract updates the pausing information whenever the hook config is modified, but this can be undetected if routed through the EVC.

Technical Details

The `adminCall()` function can be used to issue governor calls to a vault. When the call matches the `setHookConfig()` selector, the implementation updates the cached hook configuration.

However, if the transaction is routed through the EVC, then the `setHookConfig()` function selector will be nested within the calldata, without being detected at the top-level selector, which will correspond to a function of the EVC.

Impact

Low. The cached pause information may not be in sync with the actual hook configuration stored in the vault.

Recommendation

It would be difficult to parse the many potential variants of the calldata in order to extract a nested selector. As this scenario is unlikely, it is recommended to document this behavior to prevent hook modifications from being routed through the EVC.

Developer Response

We acknowledge the possibility that the `setHookConfig` call might go undetected by the `adminCall` function. A sufficient comment has been added to warn the governor admin about this possibility: https://github.com/euler-xyz/evk-periphery/pull/48/commits/8192527e9e420a313c88a2ce7ff010a7551985be

2. Low - Pausing can silently fail if guardian is not in a pausable state

Both guardian contracts simply ignore pause requests if the current state doesn’t allow it, which can be interpreted as a successful action.

Technical Details

The GovernorGuardian.sol and HookTargetGuardian.sol contracts follow a similar logic, where a pause action is ignored if the current state is not pausable.

```
    function pause() external onlyRole(GUARDIAN_ROLE) {
        if (!canBePaused()) return;
    function pause(address[] calldata vaults) external onlyRole(GUARDIAN_ROLE) {
        for (uint256 i = 0; i < vaults.length; ++i) {
            address vault = vaults[i];

            if (!canBePaused(vault)) continue;
```

A non-reverting result could be interpreted as a successful action. It is important to note that a non-pausable state does not necessarily mean the contract is currently paused. This is because there is a cooldown period, during which the contract is already unpaused but still in a non-pausable state.

Impact

Low. A non-reverting pause call may be interpreted as a successful action.

Recommendation

Consider reverting for these cases, in particular if the contract is under an unpaused cooldown state.

Developer Response

We acknowledge this behavior and would like to keep it as is. Considering that the most common use case for the `GovernorGuardian` and the `HookTargetGuardian` will involve the usage of automated systems, we would like to ensure that emergency pause transactions do not revert unnecessarily for the best possible outcome. Note that there might be multiple systems with `GUARDIAN_ROLE` granted working in parallel. In case they front-run each other and the subsets of vaults to be paused partially overlap, we would like the second transaction to be executed without reverting. Otherwise, the subset of vaults not contained in the transaction that was included first might be left unpaused and exposed to potential danger.

Informational Findings

1. Informational - Missing access control enumeration in HookTargetGuardian.sol

The HookTargetGuardian.sol contract does not implement the enumeration variant of the AccessControl library, unlike other similar contracts.

Technical Details

HookTargetGuardian.sol#L14

```
contract HookTargetGuardian is IHookTarget, AccessControl {
```

Impact

Informational.

Recommendation

Consider switching to `AccessControlEnumerable`.

Developer Response

We have changed the inherited contract from `AccessControl` to `AccessControlEnumerable` to keep the code consistent: https://github.com/euler-xyz/evk-periphery/pull/48/commits/bb8fe53928eb39ac15287b4cca2268d00e24171c

2. Informational - Pausing before unpausing will clear cached configuration

In the unlikely event a vault is paused twice before explicitly unpaused, the cached hook configuration will be overwritten with the *paused* state.

Technical Details

When a vault is first paused using GovernorGuardian.sol, its current hook configuration is stored in the `pauseDatas` mapping before setting the *paused* state so that it can be later recovered when unpausing.

```
            // Cache the hook configuration.
            (pauseDatas[vault].hookTarget, pauseDatas[vault].hookedOps) = IEVault(vault).hookConfig();
            pauseDatas[vault].lastPauseTimestamp = uint48(block.timestamp);

            // Disable all operations.
            IEVault(vault).setHookConfig(address(0), (OP_MAX_VALUE - 1));
```

If the `pause()` function is called again after the cooldown period has elapsed but before the vault is unpaused, the cached hook configuration will be overwritten with the current settings, which are `address(0)` and `OP_MAX_VALUE - 1`.

Impact

Informational.

Recommendation

The `pause()` function could be improved by relying on a boolean variable to check if the vault is currently paused, independent of the cooldown period. Otherwise, we recommend documenting this behavior.

Developer Response

Fixed as recommended: https://github.com/euler-xyz/evk-periphery/pull/48/commits/47eefc654133b766acc70c4917d28a63aefad687

Additional note

A low impact issue was found in the fix (commit 47eefc654133b766acc70c4917d28a63aefad687) where the if statement to conditionally cache the config came after the statement to pause the vault, so the guard condition would always evaluate to false and never cache the values. This issue was fixed by the team in commit 31710fe9415d97a011b469f0dbeb31bca3f3294a.

3. Informational - Selector clashing in ReadOnlyProxy.sol

A function call to the underlying implementation can be accidentally picked by the proxy if they have the same selector.

Technical Details

The ReadOnlyProxy.sol exposes two functions in addition to its fallback, `roProxyDelegateView()` and `roProxyImplementation()`.

If any of the functions present in the implementation being proxied happens to match the selector of any of these two function, then the call will be handled by the proxy instead of being dispatched to the implementation contract.

Impact

Informational.

Recommendation

The probability of a selector clash is low. However, we recommend manually verifying that the selector space intersection between the proxy and the implementation is empty. Currently, there is no clash with the functions in EVault.sol.

Developer Response

We acknowledge the possibility of a collision. We have added explicit comments in code in: https://github.com/euler-xyz/evk-periphery/pull/48/commits/90d6206882f86a0de210e9d2b2b98af63773044d

4. Informational - Make use of custom errors where possible

Technical Details

Contracts use string errors in some places, which custom errors could easily be replaced.

Impact

Informational

Recommendation

E.g., in ReadOnlyProxy.sol, replace the following reverts:

```
require(msg.sender == address(this), "unauthorized");

...

if (data.length == 0) revert("contract is in read-only mode");
```

With the following custom errors:

```
error Unauthorized();

error ReadOnlyMode();

...

if(msg.sender != address(this)) revert Unauthorized();

...

if (data.length == 0) revert ReadOnlyMode();
```

Developer Response

When pausing occurs, the read only proxy will not be verified immediately in block explorers. Reverting state modifying operations with a string message should be immediately readable by affected users. We prefer to keep as is.

5. Informational - Add view to get remaining pause time

Add a view to easily get the remaining pause time in GovernorGuardian.sol and HookTargetGuardian.sol; this view can be made public so it can be accessed by internal functions that use this and also make it easier to access this value externally.

Technical Details

Add something like:

```
function remainingPauseDuration() public view returns (uint256) {
    if (!isPaused()) return 0;
    uint256 endTime = _lastPauseTimestamp + PAUSE_DURATION;
    return endTime > block.timestamp ? endTime - block.timestamp : 0;
}
```

Impact

Informational

Recommendation

Consider adding a view to get the remaining pause duration.

Developer Response

Fixed as recommended: https://github.com/euler-xyz/evk-periphery/pull/48/commits/60b0d46aebb2633546f68e4fd993e29516870bf4

6. Informational - Additional logging

The contracts could use some additional logging, and some logs could be modified to be more informative.

Technical Details

- FactoryGovernor.pauseFactory can emit an event specifying the deployment of the new proxy (might even consider moving the event from the proxy constructor and adding the data to one event in the FactoryGovernor)

- GovernorGuardian:adminCall - it could be worth logging these, e.g., `LogAdminCall(vault, data)`. Calls requiring the guardian role, e.g., pause/unpause, could include the guardian's address triggering the event to simplifying data parsing.

Impact

Informational

Recommendation

Consider adding/modifying some events to simplify data parsing for the contracts.

Developer Response

We have added the address of the read-only proxy to the `Paused` event in the `FactoryGovernor` in: https://github.com/euler-xyz/evk-periphery/pull/48/commits/f07f8e77fdec8a81610c1d6b75a866054cfcfe27 We have also added the guardian address to the `Paused`/`Unpaused`/`PauseStatusChanged` events in the `GovernorGuardian` and the `GovernorGuardian`. Finally, we have added a new `AdminCall` event to both the `FactoryGovernor` and the `GovernorGuardian`. Both changes were implemented in: https://github.com/euler-xyz/evk-periphery/pull/48/commits/996802901fa675e3ad2005fcf394fd1c182d12fc

Final remarks

The Guardians system implements a set of contracts that serve as emergency safeguards across various protocol levels. Designed to accommodate diverse use-cases while maintaining robust security measures, this system provides different Guardian implementations with varying levels of control and intervention capabilities, from pausing to halting operations across multiple vaults simultaneously. Significant effort has been invested in designing this framework to allocate limited responsibilities to privileged roles while ensuring that access to underlying systems can be restored following an emergency intervention. This implementation enhances the overall security posture of the EVK by providing critical safeguards. The audit revealed no major issues, and the development team addressed any raised concerns promptly.