MetaMask Delegation Framework

Date	October 2024

1 Executive Summary

This report presents the results of our engagement with **MetaMask** to review **MetaMask Delegation Framework**.

The review was conducted over two weeks, from **October 21, 2024** to **October 25, 2024**, by **Rai Yang** and **Sergii Kravchenko**. A total of 5 person-days were spent.

The review is performed on the changes in the codebase that were made since the previous audit. The main changes are:

- Adding four new enforcer contracts.
- Switching the cryptography lib from Fresh Crypto Lib (FCL) to SmoothCryptoLib (SCL).
- Adding `beforeAllHook` / `afterAllHook` functions to `ICaveatEnforcer`.
- Changing order of `afterHook` executions.
- Using `EIP712` standard in the `DeleGatorCore`.

2 Scope

Our review focused on the commit hash ec0c0d64a4fc1ccca24d5e910d5712e62d84c4b7. The list of files in scope can be found in the Appendix.

2.1 Objectives

Together with the **MetaMask** team, we identified the following priorities for our review:

1. Correctness of the implementation, consistent with the intended functionality and without unintended edge cases.
2. Identify known vulnerabilities particular to smart contract systems, as outlined in our Smart Contract Best Practices, and the Smart Contract Weakness Classification Registry.

3 Findings

Each issue has an assigned severity:

- Minor issues are subjective in nature. They are typically suggestions around best practices or readability. Code maintainers should use their own judgment as to whether to address such issues.
- Medium issues are objective in nature but are not security vulnerabilities. These should be addressed unless there is a clear reason not to.
- Major issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- Critical issues are directly exploitable security vulnerabilities that need to be fixed.

3.1 Potential Misuse of `ERC1155BalanceGteEnforcer` and `ERC721BalanceGteEnforcer`. Minor

Acknowledged

Description

Two similar enforcers, `ERC1155BalanceGteEnforcer` and `ERC721BalanceGteEnforcer`, are designed to ensure that the balance increases after the transaction for no less than a specific amount.

src/enforcers/ERC721BalanceGteEnforcer.sol:L64-L71

```
{
    (address token_, address recipient_,) = getTermsInfo(_terms);
    bytes32 hashKey_ = _getHashKey(msg.sender, token_, recipient_, _delegationHash);
    require(!isLocked[hashKey_], "ERC721BalanceGteEnforcer:enforcer-is-locked");
    isLocked[hashKey_] = true;
    uint256 balance_ = IERC721(token_).balanceOf(recipient_);
    balanceCache[hashKey_] = balance_;
}
```

This pattern of checking the balance before and after the transaction can be dangerous if there is any re-entrancy possible in between. Usually, re-entrancy can trigger a separate execution flow before the previous one is finished. This separate call can also change the token balance of the target address but in an unexpected way. It can be a problem here as a generic call to another contract is happening, and there can be more executions involving the target address, potentially changing its balance. Since this is a general-purpose system, infinite scenarios can be played here, and users should be extra cautious when relying only on the balance change.

3.2 Improper Calldata Length Check in ERC721TransferEnforcer and OwnershipTransferEnforcer Contracts Minor ✓ Fixed

Resolution
Fixed.

Description

In both the `beforeHook` function of the `ERC721TransferEnforcer` contract and the `_validateAndEnforce` function of the `OwnershipTransferEnforcer` contract, the length of the `calldata` is improperly validated. The length check is performed after an operation is executed on the `calldata`, which poses a security risk. In the `beforeHook` function of the `ERC721TransferEnforcer` contract, the following line extracts the first 4 bytes of the `calldata` to determine the function selector: `bytes4 selector_ = bytes4(callData_[0:4]);` However, this operation is performed before validating that the `calldata` is at least 4 bytes long. If the `calldata` is less than 4 bytes, the contract will attempt to access out-of-bounds data, causing the transaction to revert unexpectedly. A similar issue exists in the `_validateAndEnforce` function of the `OwnershipTransferEnforcer` contract. In both cases, the `calldata` length should be checked before any operations are performed on the `calldata` to avoid unexpected reverts and ensure proper handling of `calldata` input.

Examples

src/enforcers/ERC721TransferEnforcer.sol:L36-L42

```
bytes4 selector_ = bytes4(callData_[0:4]);

// Decode the remaining callData into NFT transfer parameters
// The calldata should be at least 100 bytes (4 bytes for the selector + 96 bytes for the parameters)
if (callData_.length < 100) {
    revert("ERC721TransferEnforcer:invalid-calldata-length");
}
```

src/enforcers/OwnershipTransferEnforcer.sol:L75-L78

```
bytes4 selector = bytes4(callData_[0:4]);
require(selector == IERC173.transferOwnership.selector, "OwnershipTransferEnforcer:invalid-method");

require(callData_.length == 36, "OwnershipTransferEnforcer:invalid-execution-length");
```

Recommendation

Move the length check of `calldata` to the beginning of the function to ensure the `calldata` has sufficient length before any slicing or access operations are performed.

Appendix 1 - Files in Scope

This audit covered the following files:

File	SHA-1 hash
src/DeleGatorCore.sol	`5310e3469b74493a391cedbaec6d4f86d84c25b5`
src/DelegationManager.sol	`13c4ab79238da9107eb2f68c4bad89393f28abf8`
src/HybridDeleGator.sol	`1b34393e16b3b895ad3206511755c5a0dcbf2a34`
src/MultiSigDeleGator.sol	`aaa79cc10c8baa1820c6da7cf93f780d239c7663`
src/enforcers/CaveatEnforcer.sol	`22875c3279162b563d5767e1d2b3933f1e5bbe93`
src/enforcers/ERC1155BalanceGteEnforcer.sol	`b5c26a8dd6f53b28d07d7909d953da05529dfd3f`
src/enforcers/ERC721BalanceGteEnforcer.sol	`9e85500e3c3cf487505847c7cfbbc0c6c6b14463`
src/enforcers/ERC721TransferEnforcer.sol	`58d3e810196287094ad626067c483cb2196ae709`
src/enforcers/OwnershipTransferEnforcer.sol	`636a4066dc39f105e0626b6ccb8a0856ba046861`
src/interfaces/ICaveatEnforcer.sol	`83fbc7bfc9b2990e6f7a58f140e2b6ad40affeba`
src/interfaces/IDelegationManager.sol	`cf9b9e2c8cbac2714e37751ebe6abb10cb1ceb0b`
src/libraries/P256SCLVerifierLib.sol	`8293c94ff45c0f1b2920fcd7d2f6757e3793dda4`
src/libraries/P256VerifierLib.sol	`6366362d9224f0667305d26998221c3aa4f7b8f1`
src/libraries/WebAuthn.sol	`b10fb0bedf6ad431a17112c50debc4f687bea98e`

Appendix 2 - Disclosure

Consensys Diligence (“CD”) typically receives compensation from one or more clients (the “Clients”) for performing the analysis contained in these reports (the “Reports”). The Reports may be distributed through other means, including via Consensys publications and other distributions.

The Reports are not an endorsement or indictment of any particular project or team, and the Reports do not guarantee the security of any particular project. This Report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens

are emergent technologies and carry with them high levels of technical risk and uncertainty. No Report provides any warranty or representation to any third party in any respect, including regarding the bug-free nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on the Reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this Report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. CD owes no duty to any third party by virtue of publishing these Reports.

A.2.1 Purpose of Reports

The Reports and the analysis described therein are created solely for Clients and published with their consent. The scope of our review is limited to a review of code and only the code we note as being within the scope of our review within this report. Any Solidity code itself presents unique and unquantifiable risks as the Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond specified code that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. In some instances, we may perform penetration testing or infrastructure assessments depending on the scope of the particular engagement.

CD makes the Reports available to parties other than the Clients (i.e., “third parties”) on its website. CD hopes that by making these analyses publicly available, it can help the blockchain ecosystem develop technical best practices in this rapidly evolving area of innovation.

A.2.2 Links to Other Web Sites from This Web Site

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Consensys and CD. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Consensys and CD are not responsible for the content or operation of such Web sites, and that Consensys and CD shall have no liability to you or any other person or entity for the use of third party Web sites. Except as described below, a hyperlink from this web Site to another web site does not imply or mean that Consensys and CD endorses the content on that Web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the Reports. Consensys and CD assumes no responsibility for the use of third-party software on the Web Site and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

A.2.3 Timeliness of Content

The content contained in the Reports is current as of the date appearing on the Report and is subject to change without notice unless indicated otherwise, by Consensys and CD.