HALBORN

Animoca Brands - MetaTX & ERC20 Token

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 24th, 2022 - November 11th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	11/08/2022	Luis Arroyo
0.2	Draft Review	11/09/2022	Kubilay Onur Gungor
0.3	Draft Review	11/10/2022	Gabi Urrutia
1.0	Remediation Plan	11/22/2022	Luis Arroyo
1.1	Remediation Plan Review	12/02/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Kubilay Onur Gungor	Halborn	Kubilay.Gungor@halborn.com
Luis Arroyo	Halborn	Luis.Arroyo@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Animoca Brands engaged Halborn to conduct a security audit on their MetaTX and ERC20 Token smart contracts beginning on October 24th, 2022 and ending on November 11th, 2022. The security assessment was scoped to the MetaTX and ERC20 Token smart contracts provided in the GitHub repository animoca/ethereum-contracts with commit ID 4b4cf4535be8367ef67b2827b32ea48a2d70e79c.

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues within the smart contracts

In summary, Halborn identified some security risks that were acknowledged by the Animoca Brands team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the code and can quickly identify

items that do not follow the security best practices. The following phases and associated tools were used during the audit:

- Research into architecture and purpose
- Smart contract manual code review and walkthrough
- Graphing out functionality and contract logic/connectivity/functions (solgraph)
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes
- Manual testing by custom scripts
- Scanning of solidity files for vulnerabilities, security hotspots or bugs. (MythX)
- Static Analysis of security for scoped contract, and imported functions. (Slither)
- Testnet deployment (Brownie, Remix IDE)

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to the following smart contracts:

- MetaTX:

  -ForwarderRegistry.sol
  -ForwarderRegistryContext.sol
  -ForwarderRegistryContextBase.sol
  -ForwarderRegistryContextFacet.sol
  -IERC2771.sol
  -IForwarderRegistry.sol
  -ERC2771Calldata.sol

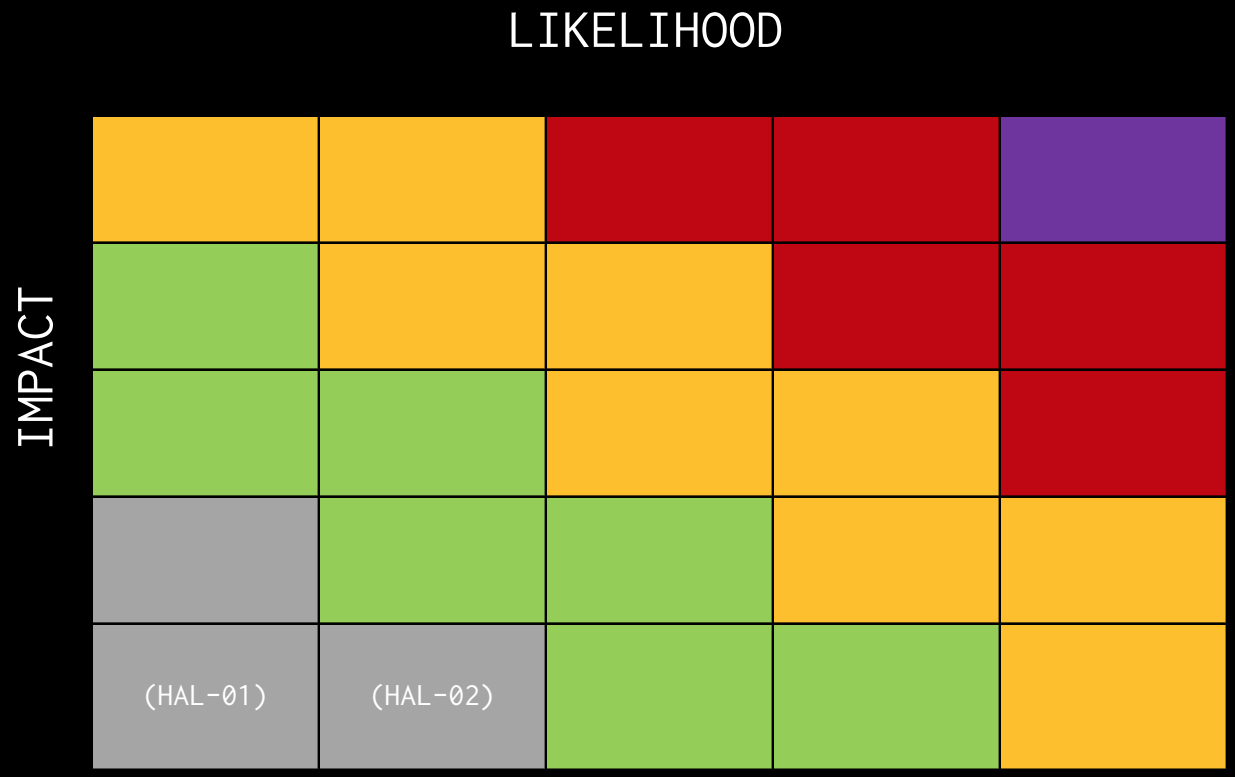
- ERC20 Token:

  -ERC20.sol
  -ERC20BatchTransfers.sol
  -ERC20Burnable.sol
  -ERC20Detailed.sol
  -ERC20Metadata.sol
  -ERC20Mintable.sol
  -ERC20Permit.sol
  -ERC20Receiver.sol
  -ERC20SafeTransfers.sol
  -ERC20Base.sol
  -ERC20BatchTransfersBase.sol
  -ERC20BurnableBase.sol
  -ERC20DetailedBase.sol
  -ERC20MetadataBase.sol
  -ERC20MintableBase.sol
  -ERC20PermitBase.sol
  -ERC20SafeTransfersBase.sol
  -ERC20BatchTransfersFacet.sol
  -ERC20BurnableFacet.sol
  -ERC20DetailedFacet.sol
  -ERC20Facet.sol

-ERC20MetadataFacet.sol
-ERC20MintableFacet.sol
-ERC20PermitFacet.sol
-ERC20SafeTransfersFacet.sol
-IERC20.sol
-IERC20Allowance.sol
-IERC20BatchTransfers.sol
-IERC20Burnable.sol
-IERC20Detailed.sol
-IERC20Metadata.sol
-IERC20Mintable.sol
-IERC20Permit.sol
-IERC20Receiver.sol
-IERC20SafeTransfers.sol
-ERC20DetailedStorage.sol
-ERC20MetadataStorage.sol
-ERC20PermitStorage.sol
-ERC20Storage.sol

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	0	2



EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - POSSIBLE OVERFLOW DUE TO NONCE DOWNCASTING	Informational	ACKNOWLEDGED
HAL02 - DIFFERENT PRAGMA VERSIONS USED	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) POSSIBLE OVERFLOW DUE TO NONCE DOWNCASTING - INFORMATIONAL

Description:

The ForwarderRegistry smart contract uses an internal struct called Forwarder, which contains the nonce, and if the address is approved as a forwarder, this nonce is declared as uint248 but the nonce received as a parameter in some functions like _setForwarderApproval() is uint256. In this function, the nonce parameter is downcasted to fit in the Forwarder struct, but this casting could occasionally create an overflow.

Code Location:

Listing 1: ForwarderRegistry.sol (Lines 169,173)

```
164 function _setForwarderApproval(
165         Forwarder storage forwarderData,
166         address sender,
167         address forwarder,
168         bool approved,
169         uint256 nonce
170     ) private {
171         forwarderData.approved = approved;
172         unchecked {
173             forwarderData.nonce = uint248(nonce + 1);
174         }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Solidity 0.8 introduces type checking for arithmetic operations, but not for type casting. It is recommended to change the nonce type in the Forwarder struct to uint256 to avoid these possible overflows.

Remediation Plan:

ACKNOWLEDGED: The Animoca Brands team added the following note about this issue:

“This scenario cannot happen. It would require that a single account sets or unsets an approval max(uint248) times, calling a SSTORE operation every time. Both the time and amount of tokens it would require are impossible to reach in any configuration of an EVM blockchain.”.

3.2 (HAL-02) DIFFERENT PRAGMA VERSIONS USED - INFORMATIONAL

Description:

The metatx and ERC20 smart contracts use floating pragma and different pragma versions, like 0.8.17. The latest pragma version, 0.8.17, released on September 8, 2022, is used in some contracts, but ^0.8.8 is also used. Avoid using floating pragmas is recommended as best practice.

Reference: Solidity Releases

Code Location:

Listing 2

```
- 0.8.17 (contracts/metatx/ForwarderRegistry.sol#2)
- ^0.8.8 (contracts/metatx/ForwarderRegistryContext.sol#2)
- ^0.8.8 (contracts/metatx/base/ForwarderRegistryContextBase.sol
↳ #2)
- 0.8.17 (contracts/metatx/facets/ForwarderRegistryContextFacet.
↳ sol#2)
- ^0.8.8 (contracts/metatx/interfaces/IERC2771.sol#2)
- ^0.8.8 (contracts/metatx/interfaces/IForwarderRegistry.sol#2)
- ^0.8.8 (contracts/metatx/libraries/ERC2771Calldata.sol#2)

- ^0.8.8 (contracts/token/ERC20/ERC20.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20BatchTransfers.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Burnable.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Detailed.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Metadata.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Mintable.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Permit.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20Receiver.sol#2)
- ^0.8.8 (contracts/token/ERC20/ERC20SafeTransfers.sol#2)
- ^0.8.8 (contracts/token/ERC20/base/ERC20Base.sol#2)
- ^0.8.8 (contracts/token/ERC20/base/ERC20BatchTransfersBase.sol
↳ #2)
- ^0.8.8 (contracts/token/ERC20/base/ERC20BurnableBase.sol#2)
- ^0.8.8 (contracts/token/ERC20/base/ERC20DetailedBase.sol#2)
```

```
22 - ^0.8.8 (contracts/token/ERC20/base/ERC20MetadataBase.sol#2)
23 - ^0.8.8 (contracts/token/ERC20/base/ERC20MintableBase.sol#2)
24 - ^0.8.8 (contracts/token/ERC20/base/ERC20PermitBase.sol#2)
25 - ^0.8.8 (contracts/token/ERC20/base/ERC20SafeTransfersBase.sol#2)
26 - 0.8.17 (contracts/token/ERC20/facets/ERC20BatchTransfersFacet.
↳ sol#2)
27 - 0.8.17 (contracts/token/ERC20/facets/ERC20BurnableFacet.sol#2)
28 - 0.8.17 (contracts/token/ERC20/facets/ERC20DetailedFacet.sol#2)
29 - 0.8.17 (contracts/token/ERC20/facets/ERC20Facet.sol#2)
30 - 0.8.17 (contracts/token/ERC20/facets/ERC20MetadataFacet.sol#2)
31 - 0.8.17 (contracts/token/ERC20/facets/ERC20MintableFacet.sol#2)
32 - 0.8.17 (contracts/token/ERC20/facets/ERC20PermitFacet.sol#2)
33 - 0.8.17 (contracts/token/ERC20/facets/ERC20SafeTransfersFacet.sol
↳ #2)
34 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20.sol#2)
35 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Allowance.sol#2)
36 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20BatchTransfers.
↳ sol#2)
37 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Burnable.sol#2)
38 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Detailed.sol#2)
39 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Metadata.sol#2)
40 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Mintable.sol#2)
41 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Permit.sol#2)
42 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20Receiver.sol#2)
43 - ^0.8.8 (contracts/token/ERC20/interfaces/IERC20SafeTransfers.sol
↳ #2)
44 - ^0.8.8 (contracts/token/ERC20/libraries/ERC20DetailedStorage.sol
↳ #2)
45 - ^0.8.8 (contracts/token/ERC20/libraries/ERC20MetadataStorage.sol
↳ #2)
46 - ^0.8.8 (contracts/token/ERC20/libraries/ERC20PermitStorage.sol
↳ #2)
```

Risk Level:

Likelihood - 2

Impact - 1

Recommendation:

It is recommended to update the pragma versions used in the metatx and ERC20 smart contracts and lock all of them on the used version 0.8.17.

Remediation Plan:

ACKNOWLEDGED: The Animoca Brands team added the following note about this issue:

“The design is intended. As a library, the project is open to different compilation options when re-usable contracts are concerned (abstract contracts, libraries, interfaces). The solidity version is the minimum version which allows to compile and passes the tests. The concrete contracts are designed to be directly deployed, in this case the latest solidity version is enforced.”.

GAS OPTIMIZATION

4.1 CHANGE THE VISIBILITY OF A FUNCTION

Description:

The ForwarderRegistry.sol contract contains a function that could be marked as external instead of public to save gas.

Listing 3: ForwarderRegistry.sol (Line 76)

```
70     function setForwarderApproval(
71         address sender,
72         address forwarder,
73         bool approved,
74         bytes calldata signature,
75         bool isEIP1271Signature
76     ) public {
```

Risk Level:

Likelihood - 2
Impact - 1

Recommendation:

It is recommended to change the public visibility of the mentioned function to external in order to save gas.

Remediation Plan:

The Animoca Brands team added the following note about this issue:

“setForwarderApproval is used in another function of the contract, which is why it is public and not external. Redesigning the code to make it external would not provide runtime gas optimization.”.

MANUAL TESTING

MANUAL TESTING

Halborn performed several manual tests in the following contracts:

- ForwarderRegistry.sol
- ForwarderRegistryContext.sol
- ERC20.sol
- ERC20Base.sol
- ERC20BatchTransfersBase.sol
- ERC20BurnableBase.sol
- ERC20DetailedBase.sol
- ERC20MetadataBase.sol
- ERC20MintableBase.sol
- ERC20PermitBase.sol
- ERC20SafeTransfersBase.sol
- ERC20DetailedStorage.sol
- ERC20MetadataStorage.sol
- ERC20PermitStorage.sol
- ERC20Storage.sol

The manual tests were focused on testing the main functions of these contracts:

- setForwarderApproval()
- forward()
- approveAndForward()
- _requireValidSignature()
- _setForwarderApproval()

MANUAL TESTING

- approve()
- increaseAllowance()
- decreaseAllowance()
- transfer()
- transferFrom()
- batchTransfer()
- batchTransferFrom()
- safeTransfer()
- safeTransferFrom()
- mint()
- batchMint()
- burn()
- burnFrom()
- batchBurnFrom()

```
  FR tests
bob is approved?:  false
bob is approved now?:  true
    ✔ setForwarderApproval
    1) forward with wrong data
    ✔ forward ok
    2) forward with wrong signature
    3) approveAndForward with wrong signature
    4) approveAndForward with wrong data
    ✔ approveAndForward ok

  testing erc20 token
bob is approved?:  false
bob is approved now?:  true
    ✔ approve
prev allowance: 0
post allowance: 100
    ✔ increaseAllowance
prev allowance: 100
post allowance: 0
    ✔ decreaseAllowance
    5) transfer without allowance
    6) transferFrom without allowance
    7) transfer err1
    8) transfer err2
    ✔ transfer
    9) transferFrom err1
    10) transferFrom err1
    ✔ transferFrom
    ✔ minting
    11) minting too much
    ✔ burning
    12) burning too much
```

```
    at ForwarderRegistry._requireValidSignature (contracts/metatx/ForwarderRegistry.sol:160)
    at ForwarderRegistry.setForwarderApproval (contracts/metatx/ForwarderRegistry.sol:80)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:135)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:136)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:119)
    at ForwarderMock.forward (contracts/mocks/metatx/ForwarderMock.sol:20)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
    at HardhatNode._mineBlockWithPendingTxs (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:1815:23)
    at HardhatNode.mineBlock (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:504:16)
    at EthModule._sendTransactionAndReturnHash (node_modules/hardhat/src/internal/hardhat-network/provider/modules/eth.ts:
:18)
    at HardhatNetworkProvider.request (node_modules/hardhat/src/internal/hardhat-network/provider/provider.ts:118:18)
    at EthersProviderWrapper.send (node_modules/hardhat-deploy-ethers/src/internal/ethers-provider-wrapper.ts:13:20)


  2) Meta Transactions
       FR tests
         forward with wrong signature:
     Error: VM Exception while processing transaction: reverted with custom error 'WrongSigner()'
    at ForwarderRegistry._requireValidSignature (contracts/metatx/ForwarderRegistry.sol:160)
    at ForwarderRegistry.setForwarderApproval (contracts/metatx/ForwarderRegistry.sol:80)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:135)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:136)
    at ForwarderMock.functionCallWithValue (@openzeppelin/contracts/utils/Address.sol:119)
    at ForwarderMock.forward (contracts/mocks/metatx/ForwarderMock.sol:20)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
    at HardhatNode._mineBlockWithPendingTxs (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:1815:23)
    at HardhatNode.mineBlock (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:504:16)
    at EthModule._sendTransactionAndReturnHash (node_modules/hardhat/src/internal/hardhat-network/provider/modules/eth.ts:
:18)
    at HardhatNetworkProvider.request (node_modules/hardhat/src/internal/hardhat-network/provider/provider.ts:118:18)
    at EthersProviderWrapper.send (node_modules/hardhat-deploy-ethers/src/internal/ethers-provider-wrapper.ts:13:20)
```

Notable case scenarios tested (i.e. prefix operator) were also discussed with Animoca Team.

Listing 4: gasTest.js

```
it("test Default", async () => {
        let tx1 = await test.testDefault(4);
        let tx2 = await tx1.wait();
        console.log('Cost:', tx2.gasUsed)
    });

    it("test Prefix", async () => {
        let tx1 = await test.testPrefix(4);
        let tx2 = await tx1.wait();
        console.log('Cost:', tx2.gasUsed)
    });
```

No issues other than those mentioned have been found during manual testing.

AUTOMATED TESTING

6.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the smart contracts in scope. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified the smart contracts in the repository and was able to compile them correctly into their ABIs and binary format, Slither was run against the contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Slither results:

Slither results for the Smart Contracts:

```
IERC20 is re-used:
        - IERC20 (contracts/token/ERC20/interfaces/IERC20.sol#7-70)
IERC20Permit is re-used:
        - IERC20Permit (contracts/token/ERC20/interfaces/IERC20Permit.sol#8-56)
ForwarderRegistry._forwarders (contracts/metatx/ForwarderRegistry.sol#32) is never initialized. It is used in:
        - ForwarderRegistry.removeForwarderApproval(address) (contracts/metatx/ForwarderRegistry.sol#56-59)
        - ForwarderRegistry.setForwarderApproval(address,address,bool,bytes,bool) (contracts/metatx/ForwarderRegistry.sol#70-82)
        - ForwarderRegistry.forward(address,bytes) (contracts/metatx/ForwarderRegistry.sol#88-92)
        - ForwarderRegistry.getNonce(address,address) (contracts/metatx/ForwarderRegistry.sol#130-132)
        - ForwarderRegistry.isApprovedForwarder(address,address) (contracts/metatx/ForwarderRegistry.sol#135-137)
ERC20Storage.batchMint(ERC20Storage.Layout,address[],uint256[]).totalValue (contracts/token/ERC20/libraries/ERC20Storage.sol#404) is a l
ocal variable never initialized
ERC20Storage.batchTransfer(ERC20Storage.Layout,address,address[],uint256[]).i (contracts/token/ERC20/libraries/ERC20Storage.sol#216) is
a local variable never initialized
TokenRecoveryBase.recoverERC20s(address[],IERC20[],uint256[]).i (contracts/security/base/TokenRecoveryBase.sol#58) is a local variable n
ever initialized
ERC20Storage.batchTransferFrom(ERC20Storage.Layout,address,address,address[],uint256[]).i (contracts/token/ERC20/libraries/ERC20Storage.
sol#272) is a local variable never initialized
ERC20Storage.batchBurnFrom(ERC20Storage.Layout,address,address[],uint256[]).i (contracts/token/ERC20/libraries/ERC20Storage.sol#499) is
a local variable never initialized
ERC20Storage.batchMint(ERC20Storage.Layout,address[],uint256[]).i (contracts/token/ERC20/libraries/ERC20Storage.sol#406) is a local vari
able never initialized
ERC20Storage.batchBurnFrom(ERC20Storage.Layout,address,address[],uint256[]).totalValue (contracts/token/ERC20/libraries/ERC20Storage.sol
#497) is a local variable never initialized
ERC20Storage.batchTransfer(ERC20Storage.Layout,address,address[],uint256[]).totalValue (contracts/token/ERC20/libraries/ERC20Storage.sol
#213) is a local variable never initialized
ERC20Storage.batchTransfer(ERC20Storage.Layout,address,address[],uint256[]).selfTransferTotalValue (contracts/token/ERC20/libraries/ERC2
0Storage.sol#214) is a local variable never initialized
ERC20Storage.batchTransferFrom(ERC20Storage.Layout,address,address,address[],uint256[]).totalValue (contracts/token/ERC20/libraries/ERC2
0Storage.sol#269) is a local variable never initialized
ERC20Storage.batchTransferFrom(ERC20Storage.Layout,address,address,address[],uint256[]).selfTransferTotalValue (contracts/token/ERC20/li
braries/ERC20Storage.sol#270) is a local variable never initialized
ForwarderRegistry.forward(address,bytes) (contracts/metatx/ForwarderRegistry.sol#88-92) ignores return value by target.functionCallWithV
alue(abi.encodePacked(data,sender),msg.value) (contracts/metatx/ForwarderRegistry.sol#91)
```

```
ForwarderRegistry.approveAndForward(bytes,bool,address,bytes) (contracts/metatx/ForwarderRegistry.sol#102-111) ignores return value by t
arget.functionCallWithValue(abi.encodePacked(data,sender),msg.value) (contracts/metatx/ForwarderRegistry.sol#110)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
        - ForwarderRegistryContext.forwarderRegistry() (contracts/metatx/ForwarderRegistryContext.sol#14-16) (function)
ERC20PermitStorage.permit(ERC20PermitStorage.Layout,address,address,uint256,uint256,uint8,bytes32,bytes32) (contracts/token/ERC20/librar
ies/ERC20PermitStorage.sol#41-60) uses timestamp for comparisons
        - require(bool,string)(block.timestamp <= deadline,ERC20: expired permit) (contracts/token/ERC20/libraries/ERC20PermitStorage.so
l#52)
```

```
Pragma version0.8.17 (contracts/metatx/ForwarderRegistry.sol#2) necessitates a version too recent to be trusted. Consider deploying with
 0.6.12/0.7.6/0.8.7
Pragma version^0.8.8 (contracts/metatx/ForwarderRegistryContext.sol#2) is known to contain severe issues (https://solidity.readthedocs.i
o/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/metatx/base/ForwarderRegistryContextBase.sol#2) is known to contain severe issues (https://solidity.read
thedocs.io/en/latest/bugs.html)
Pragma version0.8.17 (contracts/metatx/facets/ForwarderRegistryContextFacet.sol#2) necessitates a version too recent to be trusted. Cons
ider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.8 (contracts/metatx/interfaces/IERC2771.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/
latest/bugs.html)
Pragma version^0.8.8 (contracts/metatx/interfaces/IForwarderRegistry.sol#2) is known to contain severe issues (https://solidity.readthed
ocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/metatx/libraries/ERC2771Calldata.sol#2) is known to contain severe issues (https://solidity.readthedocs.
io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/latest/bu
gs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20BatchTransfers.sol#2) is known to contain severe issues (https://solidity.readthedocs.i
o/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Burnable.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/l
atest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Detailed.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/l
atest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Metadata.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/l
atest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Mintable.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/l
atest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Permit.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/lat
est/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/ERC20Receiver.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/l
atest/bugs.html)
```

```
Pragma version^0.8.8 (contracts/token/ERC20/ERC20SafeTransfers.sol#2) is known to contain severe issues (https://solidity.readthedocs.io
/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20Base.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/en/
latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20BatchTransfersBase.sol#2) is known to contain severe issues (https://solidity.read
thedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20BurnableBase.sol#2) is known to contain severe issues (https://solidity.readthedoc
s.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20DetailedBase.sol#2) is known to contain severe issues (https://solidity.readthedoc
s.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20MetadataBase.sol#2) is known to contain severe issues (https://solidity.readthedoc
s.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20MintableBase.sol#2) is known to contain severe issues (https://solidity.readthedoc
s.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20PermitBase.sol#2) is known to contain severe issues (https://solidity.readthedocs.
io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/base/ERC20SafeTransfersBase.sol#2) is known to contain severe issues (https://solidity.readt
hedocs.io/en/latest/bugs.html)
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20BatchTransfersFacet.sol#2) necessitates a version too recent to be trusted. Cons
ider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20BurnableFacet.sol#2) necessitates a version too recent to be trusted. Consider d
eploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20DetailedFacet.sol#2) necessitates a version too recent to be trusted. Consider d
eploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20Facet.sol#2) necessitates a version too recent to be trusted. Consider deploying
 with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20MetadataFacet.sol#2) necessitates a version too recent to be trusted. Consider d
eploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20MintableFacet.sol#2) necessitates a version too recent to be trusted. Consider d
eploying with 0.6.12/0.7.6/0.8.7
```

AUTOMATED TESTING

```
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20PermitFacet.sol#2) necessitates a version too recent to be trusted. Consider dep
loying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/token/ERC20/facets/ERC20SafeTransfersFacet.sol#2) necessitates a version too recent to be trusted. Consi
der deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20.sol#2) is known to contain severe issues (https://solidity.readthedocs.io/
en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Allowance.sol#2) is known to contain severe issues (https://solidity.readth
edocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20BatchTransfers.sol#2) is known to contain severe issues (https://solidity.r
eadthedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Burnable.sol#2) is known to contain severe issues (https://solidity.readthe
docs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Detailed.sol#2) is known to contain severe issues (https://solidity.readthe
docs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Metadata.sol#2) is known to contain severe issues (https://solidity.readthe
docs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Mintable.sol#2) is known to contain severe issues (https://solidity.readthe
docs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Permit.sol#2) is known to contain severe issues (https://solidity.readthedo
cs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20Receiver.sol#2) is known to contain severe issues (https://solidity.readthe
docs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/interfaces/IERC20SafeTransfers.sol#2) is known to contain severe issues (https://solidity.re
adthedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/libraries/ERC20DetailedStorage.sol#2) is known to contain severe issues (https://solidity.re
adthedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/libraries/ERC20MetadataStorage.sol#2) is known to contain severe issues (https://solidity.re
adthedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/libraries/ERC20PermitStorage.sol#2) is known to contain severe issues (https://solidity.read
thedocs.io/en/latest/bugs.html)
Pragma version^0.8.8 (contracts/token/ERC20/libraries/ERC20Storage.sol#2) is known to contain severe issues (https://solidity.readthedoc
s.io/en/latest/bugs.html)
Function ForwarderRegistry.DOMAIN_SEPARATOR() (contracts/metatx/ForwarderRegistry.sol#116-124) is not in mixedCase
Variable ForwarderRegistryContextBase._forwarderRegistry (contracts/metatx/base/ForwarderRegistryContextBase.sol#11) is not in mixedCase
Function ERC20PermitBase.DOMAIN_SEPARATOR() (contracts/token/ERC20/base/ERC20PermitBase.sol#35-37) is not in mixedCase
Function ERC20PermitStorage.DOMAIN_SEPARATOR() (contracts/token/ERC20/libraries/ERC20PermitStorage.sol#87-102) is not in mixedCase
Variable ForwarderRegistryContextBase._forwarderRegistry (contracts/metatx/base/ForwarderRegistryContextBase.sol#11) is too similar to F
orwarderRegistryContext.constructor(IForwarderRegistry).forwarderRegistry_ (contracts/metatx/ForwarderRegistryContext.sol#12)
onERC20Received(address,address,uint256,bytes) should be declared external:
        - ERC20Base.allowance(address,address) (contracts/token/ERC20/base/ERC20Base.sol#60-62)
```

- No major issues found by Slither.

6.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on the smart contracts and sent the compiled results to the analyzers in order to locate any vulnerabilities.

MythX results:

MythX results for MetaTX.

Report for ForwarderRegistryContext.sol
https://dashboard.mythx.io/#/console/analyses/b4ec6fd3-5b90-4f48-a7ef-4a407f7e32bc

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for base/ForwarderRegistryContextBase.sol
https://dashboard.mythx.io/#/console/analyses/b4ec6fd3-5b90-4f48-a7ef-4a407f7e32bc

Line	SWC Title	Severity	Short Description
21	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
39	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.

MythX results for ERC20 Token.

AUTOMATED TESTING

Report for contracts/token/ERC20/ERC20.sol
https://dashboard.mythx.io/#/console/analyses/8fd95268-12e6-4a48-bc09-1a0594db4d6a

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20Base.sol
https://dashboard.mythx.io/#/console/analyses/a18914fb-524e-4939-b568-4a4fe492d53a

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/libraries/ERC20Storage.sol
https://dashboard.mythx.io/#/console/analyses/3067dd35-2231-4ae9-b2db-6f0854f250e4
https://dashboard.mythx.io/#/console/analyses/6aa8b350-81b9-4bf9-b3d7-98e4074f63c7

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
573	(SWC-123) Requirement Violation	Low	Requirement violation.

Report for contracts/token/ERC20/ERC20BatchTransfers.sol
https://dashboard.mythx.io/#/console/analyses/d3e7a015-e4b7-4023-8cf9-43b65bf6a92e

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

AUTOMATED TESTING

Report for contracts/token/ERC20/base/ERC20BatchTransfersBase.sol
https://dashboard.mythx.io/#/console/analyses/e9367350-088b-40eb-b660-00db8bc879f2

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20Burnable.sol
https://dashboard.mythx.io/#/console/analyses/50b6223d-a317-494d-833c-d089e0e44328

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20BurnableBase.sol
https://dashboard.mythx.io/#/console/analyses/68becea7-bcbd-4a67-8dbf-17d5afde8a1a

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20Detailed.sol
https://dashboard.mythx.io/#/console/analyses/6a554f87-b839-4238-9963-20ea94136b79

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20DetailedBase.sol
https://dashboard.mythx.io/#/console/analyses/f2479446-4872-41ba-90c4-7795b3edc29a

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

AUTOMATED TESTING

Report for contracts/token/ERC20/libraries/ERC20DetailedStorage.sol
https://dashboard.mythx.io/#/console/analyses/0408136e-79c1-41d8-8cd4-6f85eec5b6d8

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20Metadata.sol
https://dashboard.mythx.io/#/console/analyses/082a83d8-ab4b-4b2f-8a72-3795b9940d18

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20MetadataBase.sol
https://dashboard.mythx.io/#/console/analyses/d4f1c20a-1b3c-42e1-868c-4c606dcc5a17

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/libraries/ERC20MetadataStorage.sol
https://dashboard.mythx.io/#/console/analyses/a1c8f28e-a397-4008-8280-a3c02f5daef2

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20Mintable.sol
https://dashboard.mythx.io/#/console/analyses/babe1fc5-b7c6-4fd0-8249-79c63698f310

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20MintableBase.sol
https://dashboard.mythx.io/#/console/analyses/82ac5056-6e7c-4535-a558-473965ef5523

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20Permit.sol
https://dashboard.mythx.io/#/console/analyses/84c57c88-b7f2-4fc4-81c0-c99d12892766

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20PermitBase.sol
https://dashboard.mythx.io/#/console/analyses/b9f08534-5bb9-4968-ad85-61d7b64a4714

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/libraries/ERC20PermitStorage.sol
https://dashboard.mythx.io/#/console/analyses/f2e7ab5d-4dff-4ec1-9a62-da5853b756e6

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for token/ERC20/ERC20Receiver.sol
https://dashboard.mythx.io/#/console/analyses/37abeab2-fc98-46d8-aa72-14b78106d512

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/ERC20SafeTransfers.sol
https://dashboard.mythx.io/#/console/analyses/c35b4cd3-b071-4d31-84dd-74b75d4cfd73

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/base/ERC20SafeTransfersBase.sol
https://dashboard.mythx.io/#/console/analyses/51b64610-4ac6-4b0b-a082-e61638c93f18

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Report for contracts/token/ERC20/facets/ERC20SafeTransfersFacet.sol
https://dashboard.mythx.io/#/console/analyses/6aa8b350-81b9-4bf9-b3d7-98e4074f63c7

Line	SWC Title	Severity	Short Description
14	(SWC-123) Requirement Violation	Low	Requirement violation.

THANK YOU FOR CHOOSING

// HALBORN