information systems development

SMART CONTRACT AUDIT REPORT

Envelop V1 Protocol Core

2023-04-20, Revision 2

Subject of the audit

Envelop project repository containing **Envelop V1 Core** smart contract was provided for audit, release 1.1.0

1. https://github.com/dao-envelop/envelop-protocol-v1/commit/7899b4161d8ada56206cba6212f69d119cf156ea
2. https://github.com/dao-envelop/envelop-protocol-v1/commit/721830fd9afe62d5a7a1f7bae06e78afae6bc9f6

Core contracts:

- AdvancedWhiteList.sol
- TechTokenV1.sol
- FeeRoyaltyModelV1_00.sol
- EnvelopwNFT721.sol
- EnvelopwNFT1155.sol
- WrappedBaseV1.sol
- TokenService.sol

Report stages

Stage	Description
Revision 1	2023-04-18, At commit 7899b4161d8ada56206cba6212f69d119cf156ea
Revision 2	2023-04-20, At commit 721830fd9afe62d5a7a1f7bae06e78afae6bc9f6

Audit methodology

During the audit a manual code analysis was applied. The code was also analyzed in terms of compliance with best solidity development practice. For some cases, the Foundry unit test was implemented.

Classification of investigated vulnerabilities:

Critical

Loss funds by users in time of work of smart contracts. Presence of malicious code.

Loss of ownership of smart contracts.

Medium

All or part methods are stopping to work, but it does not lead to loss of user funds or freezing user funds during some times. Unauthorized changing of value of important smart contracts variables and mappings.

Low

The actions of a malicious user may cause unexpected contract behavior, including the loss **of that user**'s funds. For example tokens or ether sent to simple erc20 contract address will be lost if there is no special functions or fallbacks.

Minor

Other unwanted or unexpected behavior of smart contract methods.

Info

Subjective opinion of the auditor based on best practices.

Audit Issues and Notes

1. Unwanted removal of element from white list in **removeWLItem**

Location	Description	Severity	Status
AdvancedWhiteList.sol	If Owner pass wrong ETypes.Asset param in function **removeWLItem** (not existing in whitelist), then first element of whitelist (index 0) will still be deleted. https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L58 https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L90 After for loop there is not checking, if deletedIndex was updated or not. If not (default value 0), will be deleted element with index 0.	**Medium**	Resloved

Recommendation

Use additional boolean variable - bool found.

Team Feedback

Issue Accepted and will be solved in next release.
Resolved at commit:
https://github.com/dao-envelop/envelop-protocol-v1/commit/721830fd9afe62d5a7a1f7bae06e78afae6bc9f6

Audit Issues and Notes

2. No revert in **wrap** in case invalid parameters

Location	Description	Severity	Status
TokenService.sol, WrappedBaseV1.sol	if user pass invalid type of _inData.outType in function **wrap**, function will not be reverted https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/TokenService.sol#L31	Low	Resloved

Recommendation

Add block 'else' with revert UnSupportedAsset().

Team Feedback

Issue Accepted and will be solved in next release.
Resolved at commit:
https://github.com/dao-envelop/envelop-protocol-v1/commit/721830fd9afe62d5a7a1f7bae06e78afae6bc9f6

Audit Issues and Notes

3. Call **unWrap** from malicious contract

Location	Description	Severity	Status
WrappedBaseV1.sol	User could lose his native tokens, when call **unWrap** function with parameter _isEmergency=true and if the user is contract, which dont have receive() function. Description: Internal function _transferEmergency ignores sending status of native tokens and always deletes nft from user’s balance. Native tokens will be on contract’s balance forever. And even the owner could not withdraw it. https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/WrapperBaseV1.sol#L624	Low	Canceled

Recommendation

Allow users to change withdraw address, if first attempt has failed.

Team Feedback

The described situation can only arise as a result of malicious user behavior. Calling the **unWrap** function from a contract that is not intended to hold assets is not normal user scenarios. It is the same as sending tokens or ether to zero address and losing it in that case. The team proposes to cancel this issue.

Audit Issues and Notes

4. Out of gas in white and black lists loops

Location	Description	Severity	Status
AdvancedWhiteList.sol	There are 4 for loops, which use whiteListedArray and blackListedArray. If arrays have many elements, for loops could run out of gas. Recommended:. https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L28 https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L48 https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L71 https://github.com/dao-envelop/envelop-protocol-v1/blob/7899b4161d8ada56206cba6212f69d119cf156ea/contracts/AdvancedWhiteList.sol#L80	Minor	Identified

Recommendation

Use pagination method.

Team Feedback

It is necessary to distinguish between checking (1) elements for presence in the white or black list and actually editing (2) these lists. In the main user scenario, only checking(1) is used. When **checking**, out of gas cannot happen because of the mapping use (**whiteList[address], blackList[address]**). **Editing** these lists is a feature available to the owner of the contract. When editing, out of gas can happen if there are more than 1000 items in the lists. This size of the lists is not a normal contract scenario. And in such cases, the contract owner can simply disable the whitelist and blacklist feature.

The team proposes to downgrade the severity of this issue in the report.

Conclusions

During the audit some vulnerabilities were identified and reported to the team in auditor report Revision 1.

The Envelop team deeply analyzed the Revision 1 report and discussed the issues with the auditor. As a result, two vulnerabilities were fixed. One issue was downgraded and one was canceled.

There are no critical and medium vulnerabilities detected in Envelop ProtocolV1 Core contracts at the time of compiling the final (rev2) auditor's report.

Contact with us:

https://iber.group/

tech@iber.group

Auditor: Dmitriev Viacheslav , Email: slavahustle@gmail.com

2023-04-20, Revision 2