LUKSO Genesis Deposit Audit Report

Feb 9, 2023

WATCHPUG

Table of Contents

Summary

This report has been prepared for LUKSO Genesis Deposit Audit Report smart contract, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Overview

Project Summary

Project Name	**LUKSO Genesis Deposit Audit Report**
Codebase	**https://github.com/lukso-network/network-genesis-deposit-contract**
Commit	**c8a1b77aee09b12fe28e766eb2aef5d57cd14224**
Language	**Solidity**

Audit Summary

Delivery Date	**Feb 9, 2023**
Audit Methodology	**Static Analysis, Manual Review**
Total Isssues	**3**

[WP-I1] Consider adding a timelock to the `freezeContract()` function

Informational

Issue Description

https://github.com/lukso-network/network-genesis-deposit-contract/blob/c3d5ba9e48cc448bb4487603e1bfd3929d3b5dee/contracts/LUKSOGenesisValidatorsDepositContract.sol#L157-L160

```
157   function freezeContract() external {
158           require(msg.sender == owner, "LUKSOGenesisValidatorsDepositContract:
      Caller not owner");
159           isContractFrozen = true;
160       }
```

Given the severe impact of halting the main function of the contract, and the fact that it is irreversible once `freezeContract()` is called, careful consideration must be taken before making such a decision.

Recommendation

Consider adding a time lock or two-step confirmation procedure to the `freezeContract()` function.

This would help to avoid freezing the contract by mistake.

Status

✓ **Fixed**

[WP-I2] Consider revoking the operator privilege on `LUKSOGenesisValidatorsDepositContract`

Informational

Issue Description

It is safe not to fix this issue as `LYXe.defaultOperators()` is empty on production.

```
363  function isOperatorFor(
364      address operator,
365      address tokenHolder
366  ) public view returns (bool) {
367      return operator == tokenHolder ||
368          (_defaultOperators[operator] &&
     !_revokedDefaultOperators[tokenHolder][operator]) ||
369          _operators[tokenHolder][operator];
370  }
```

```
404  function operatorSend(
405      address sender,
406      address recipient,
407      uint256 amount,
408      bytes calldata data,
409      bytes calldata operatorData
410  )
411  external
412  {
413      require(isOperatorFor(msg.sender, sender), "ERC777: caller is not an operator
     for holder");
414      _send(msg.sender, sender, recipient, amount, data, operatorData, true);
415  }
```

The LYXe token allows its `defaultOperators` to transfer LYXe from any address, which includes contracts like `LUKSOGenesisValidatorsDepositContract` .

But the LYXe tokens sent to the `LUKSOGenesisValidatorsDepositContract` are not supposed to be transferred out. Therefore, the privilege of `LYXe.defaultOperators()` should be revoked on `LUKSOGenesisValidatorsDepositContract` .

For example, calling `revokeOperator()` in the constructor function on the `LUKSOMigrationDepositContract` contract:

```
address[] memory defaultOperators = ILYXe(LYXeAddress).defaultOperators();
for (uint256 i = 0; i < defaultOperators.length; i++) {
    ILYXe(LYXeAddress).revokeOperator(defaultOperators[i]);
}
```

Status

ⓘ Acknowledged

[WP-I3] Supply vote `0` should be considered as a non-vote

Informational

Issue Description

https://github.com/lukso-network/network-genesis-deposit-contract/blob/c8a1b77aee09b12fe28e766eb2aef5d57cd14224/contracts/LUKSOGenesisValidatorsDepositContract.sol#L175-L184

```
175  function getsVotesPerSupply()
176      external
177      view
178      returns (uint256[101] memory votesPerSupply, uint256 totalVotes)
179  {
180      for (uint256 i = 0; i <= 100; i++) {
181          votesPerSupply[i] = supplyVoteCounter[i];
182      }
183      return (votesPerSupply, deposit_count);
184  }
```

https://github.com/lukso-network/network-genesis-deposit-contract/blob/c8a1b77aee09b12fe28e766eb2aef5d57cd14224/contracts/LUKSOGenesisValidatorsDepositContract.sol#L138-L140

```
138  uint8 supply = uint8(depositData[208]);
139  require(supply <= 100, "LUKSOGenesisValidatorsDepositContract: Invalid supply
     vote");
140  supplyVoteCounter[supply]++;
```

According to the spec:

> for the vote feature (last byte of the DepositData byte), we are allowing as the LYX total supply, a value between 0 and 100 where 0 is considered as a non-vote.

However, the current implementation still stores and returns the 0 votes, which is unnecessary.

Even if there is a need to count the 0 votes, it can still be computed as `deposit_count - sumOf(supplyVoteCounter[1..100])` .

Recommendation

Consider changing to:

```
175  function getsVotesPerSupply()
176      external
177      view
178      returns (uint256[100] memory votesPerSupply, uint256 totalVotes)
179  {
180      for (uint256 i = 1; i <= 100; i++) {
181          votesPerSupply[i] = supplyVoteCounter[i];
182      }
183      return (votesPerSupply, deposit_count);
184  }
```

```
138  uint8 supply = uint8(depositData[208]);
139  require(supply <= 100, "LUKSOGenesisValidatorsDepositContract: Invalid supply
     vote");
140  if (supply > 0) supplyVoteCounter[supply]++;
```

Status

ⓘ Acknowledged

Appendix

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by WatchPug; however, WatchPug does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Smart Contract technology 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 the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.