# LandOrc

## Smart Contract Audit (Final Report)

1st October 2021

For: LandOrc
Prepared By: Entersoft Pte Ltd of 1B Trengganu Street, Singapore 058455

# Contents

## Revision History and Version Control

| Version | Date | Author(s) | Description |
|---|---|---|---|
| **1.0** | September 30th,2021 | ES Auditors | Initial Draft of Final Report |
| **1.0** | October 1st ,2021 | Jake Lemke | Reviewed |
| **1.0** | October 1st ,2021 | Paul Kang | Released Final Report |

Entersoft was commissioned by LandOrc to perform a source code review on their solidity smart contract. The review was conducted from August 29th, 2021 to September 3rd, 2021. The report is organized into the following sections:

- Executive Summary: A high-level overview of the security audit findings.
- Technical Analysis: Our detailed analysis of the Smart Contract code

The information in this report should be used to understand overall code quality, security, correctness while confirming that the code will work as LandOrc described in the smart contract."

# 1.0 Disclaimer

This is a limited audit report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to: (i) smart contract best coding practices and issues in the framework and algorithms based on white paper, code, the details of which are set out in this report, (Smart Contract audit). To get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us based on what it says or does not say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full. DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Entersoft Australia and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Entersoft) owe no duty of care towards you or any other person, nor does Entersoft make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Entersoft hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Entersoft hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Entersoft, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the Smart contract is purely based on the smart contract code shared with us alone.

# 2.0 Overview

## 2.1 Project Overview

During the period of **August 29th , 2021 to September 3rd , 2021**– Entersoft performed security audits for **LandOrc (LORC)** smart contracts.

## 2.2 Scope

The scope of this audit was to analyse and document the LandOrc smart contract codebase for quality, security, and correctness.

**OUT-OF-SCOPE:** External contracts, External Oracles, other smart contracts in the repository or imported smart contracts.

## 2.3 Project Summary

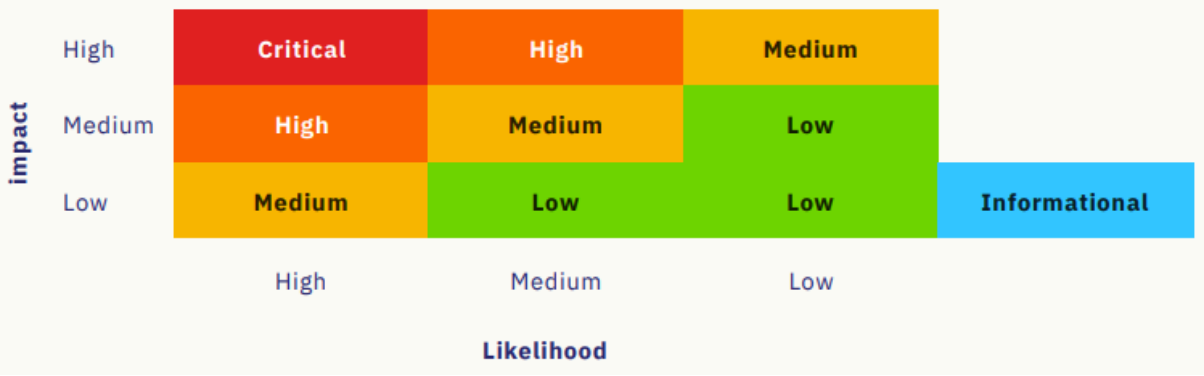
| | |
|---|---|
| **Project Name** | LandOrc |
| **Platform** | Ethereum |
| **Codebase** | https://www.dropbox.com/sh/i87av8n19un71dh/AABVps8FDbIsD3f-MUSG6C0ua?dl=0 |
| **Token Name** | N/A |
| **Contract Name(s)** | LandGov.sol, LandGovLegal.sol, LandOrc.sol, LandOrcNft.sol, LandOrcFinance.sol, LandOrcFinance.sol, LandStaking.sol |
| **Contract Address** | N/A |
| **Verified** | Yes |
| **Audited** | Yes |
| **Vulnerabilities / Issues** | Below |

## 2.4 Audit Summary

| | |
|---|---|
| **Delivery Date** | 1st October September 2021 |
| **Consultants Engaged** | 1 |

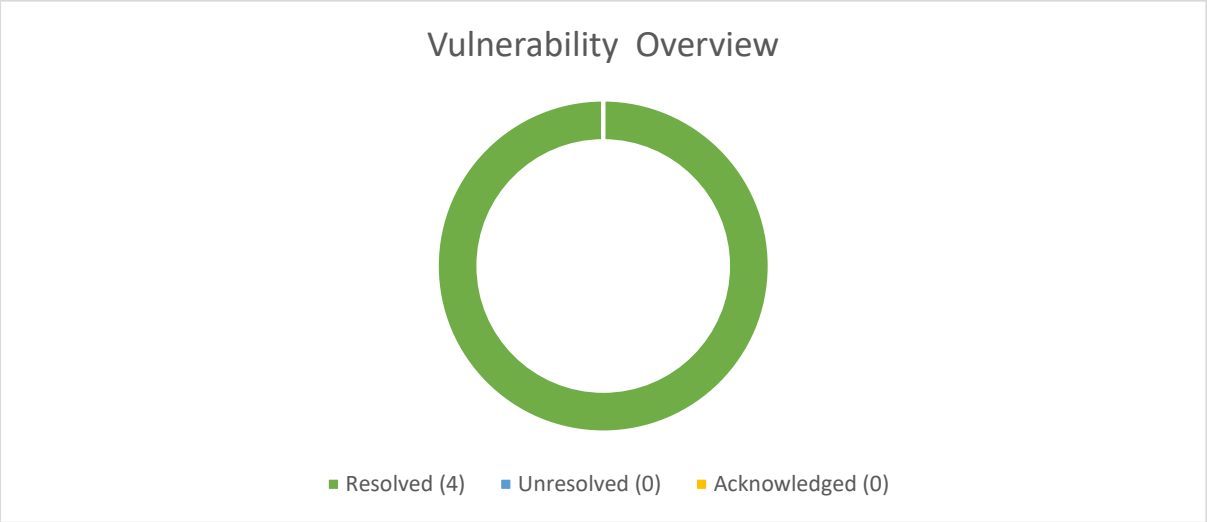
## 2.5 Security Level references

Every issue in this report was assigned a severity level from the following classification table:



## 2.6 Vulnerability Summary

| | |
|---|---|
| **Total Critical** | 0 |
| **Total High** | 0 |
| **Total Medium** | 0 |
| **Total Low** | 0 |
| **Total Informational** | 4 |




## 2.7 Audit Results Overview

| Audit Item | Audit Subclass | Audit Result |
|---|---|---|
| **Overflow** | - | Passed |
| **Race Conditions** | - | Passed |
| **Permissions** | Permission Vulnerability Audit<br>Excessive Auditing Authority | Passed |
| **Safety Design** | Zeppelin Safe Math | Passed |
| **DDOS Attack** | Call Function Security | Passed |
| **Gas Optimization** | - | Passed |
| **Design Logic** | - | Passed |
| **Know Attacks** | - | Passed |
| **Overall Audit Result** | - | Passed |

# 3.0 Executive Summary

## 3.1 Files in Scope

## 3.2 Findings

| ID | Title | Severity | Resolved |
|---|---|---|---|
| **AXVL-001** | SafeMath is not used everywhere | Informational | **Closed** |
| **AXVL-002** | Solidity Style guide is not followed | Informational | **Closed** |
| **AXVL-003** | Doesn't use SafeMath | Informational | **Closed** |
| **AXVL-004** | Unnecessary use of require statement | Informational | **Closed** |

## 3.3 Comments

Overall, the smart contracts are very well written and adhere to guidelines.

No instances of Integer Overflow and Underflow vulnerabilities or Back-Door Entry were found in the contract but relying on other contracts might cause Re-entrancy Vulnerability.

Some informational severity issues were detected; it is recommended to fix them.

# 4.0 Vulnerabilities

## 4.1 SafeMath is not used everywhere

| Severity | Confidence | Status |
|---|---|---|
| Informational | High | Closed |

**Description**

Use of Safe math is required in each operation

**Remediation**
Use sub()

## 4.2 Solidity style guide is not followed

| Severity | Confidence | Status |
|---|---|---|
| Informational | High | Closed |

**Description**

| Line | Code |
|---|---|
| LorcGeneralStaking | Inside each contract, library or interface, use the following order:<br>1. Type declarations<br>2. State variables<br>3. Events<br>4. Functions |

It might be clearer to declare types close to their use in events or state variables.

**Remediation**
Please check the order of event declaration in all the files

## 4.3 Doesn't use SafeMath

| Severity | Confidence | Status |
|---|---|---|
| Informational | High | Closed |

**Description:**

| Line | Code |
|---|---|
| LorcStaking L41 | txFeePercentage = 30; // 0.3% = 0.3/100 * 10000 |

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

**Remediation**

Use SafeMath in all operations

## 4.4 Reentrancy in unStake

| Severity | Confidence | Status |
|---|---|---|
| Informational | High | Closed |

**Description**

| Line | Code |
|---|---|
| LandGovLegal | function withdrawErc20(IERC20 token, uint amount) external onlyRole(DEFAULT_ADMIN_ROLE) {<br>require(amount > 0, "LandGovLegal: invalid amount");<br>require(token.balanceOf(address(this)) >= amount, "LandGovLegal: insufficient ERC20 balance");<br>require(token.transfer(msg.sender, amount), "LandGovLegal: transfer failed");<br>} |

require(token.balanceOf(address(this)) >= amount, "LandGovLegal: insufficient ERC20 balance");

**Remediation**

Transfer will be automatically failed if, contract doesn't have balance, gas will be consumed more if unnecessary require statement is used.

## 5.0 LandOrc Functional Tests

The following is the list of functions tested and checked for vulnerabilities during audit:

### Landgov

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| addBlackList | Pass | Pass | Pass |
| removeBlackList | Pass | Pass | Pass |
| pause | Pass | Pass | Pass |
| unpause | Pass | Pass | Pass |
| mint | Pass | Pass | Pass |
| emergencyWithdraw | Pass | Pass | Pass |

### LandGovLegal

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| setLorcFinancing | Pass | Pass | Pass |
| totalLegalTeam | Pass | Pass | Pass |
| hasVoted | Pass | Pass | Pass |
| getVoteCount | Pass | Pass | Pass |
| getMinVerifierRequired | Pass | Pass | Pass |
| verifyLorcFinancing | Pass | Pass | Pass |

### LandOrc

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| setLandNFTAddress | Pass | Pass | Pass |
| setRewardVaultAddress | Pass | Pass | Pass |
| setExchangeAddress | Pass | Pass | Pass |
| setRewardPercentage | Pass | Pass | Pass |
| addBlackList | Pass | Pass | Pass |
| removeBlackList | Pass | Pass | Pass |
| isVestingCompleted | Pass | Pass | Pass |
| addAllocations | Pass | Pass | Pass |
| addFrozenWallet | Pass | Pass | Pass |
| canTransfer | Pass | Pass | Pass |
| mintNFTReward | Pass | Pass | Pass |
| mint | Pass | Pass | Pass |

### LandOrcNft

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| safeMint | Pass | Pass | Pass |
| updateTokenURI | Pass | Pass | Pass |
| tokenURI | Pass | Pass | Pass |
| setLorcAddress | Pass | Pass | Pass |
| supportsInterface | Pass | Pass | Pass |

### Functional Tests LandOrcFinance

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| setVerifiedAddress | Pass | Pass | Pass |
| updateTxFee | Pass | Pass | Pass |
| setLandGovVotePercentage | Pass | Pass | Pass |
| getTxFee | Pass | Pass | Pass |
| getFinancingID | Pass | Pass | Pass |

### Functional Tests LandOrcFinance

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| updateMinApprovalRequired | Pass | Pass | Pass |
| addApprovers | Pass | Pass | Pass |
| confirmTransaction | Pass | Pass | Pass |
| executeTxn | Pass | Pass | Pass |
| approveTxn | Pass | Pass | Pass |
| removeApprovers | Pass | Pass | Pass |
| getTxnCount | Pass | Pass | Pass |
| submitTransaction | Pass | Pass | Pass |

### Functional Tests LandStaking

| Function Name() | Technical Result | Logical Result | Overall Result |
|---|---|---|---|
| setLorcAddress | Pass | Pass | Pass |
| updateTxFee | Pass | Pass | Pass |
| getTxFee | Pass | Pass | Pass |
| stake | Pass | Pass | Pass |
| withdrawReward | Pass | Pass | Pass |
| restakeReward | Pass | Pass | Pass |
| unstakeAll | Pass | Pass | Pass |
| distributeRewards | Pass | Pass | Pass |

# 6.0 Automated Testing

Automated testing is carried out with the following tools:

- Slither
- Mythril
- Echidna
- Manticore

## 6.1 Surya

```
+  LorcPreSale (Ownable, Pausable)
   - [Pub] <Constructor> #
   - [Ext] updateETHRate #
      - modifiers: onlyOwner
   - [Ext] updateUSDTRate #
      - modifiers: onlyOwner
   - [Ext] buyWithETH ($)
      - modifiers: whenNotPaused
   - [Ext] buyWithUSDT #
      - modifiers: whenNotPaused
   - [Ext] withdrawETH #
      - modifiers: onlyOwner
   - [Ext] withdrawErc20 #
      - modifiers: onlyOwner
   - [Ext] pause #
      - modifiers: onlyOwner,whenNotPaused
   - [Ext] unpause #
      - modifiers: onlyOwner,whenPaused
   - [Ext] <Fallback> ($)

+  LorcRewardVault (Initializable, AccessControlEnumerableUpgradeable, PausableUpgradeable, UUPSUpgradeable, DSMath)
   - [Pub] initialize #
      - modifiers: initializer
   - [Ext] name
   - [Ext] updateMinApprovalRequired #
      - modifiers: onlyRole
   - [Ext] addApprovers #
      - modifiers: onlyRole
   - [Ext] removeApprovers #
      - modifiers: onlyRole
   - [Ext] getTxnCount
   - [Ext] getTxn
   - [Ext] submitTransaction #
      - modifiers: onlyRole
   - [Ext] confirmTransaction #
      - modifiers: onlyRole
   - [Ext] executeTxn #
```

```
+  LorcFinancing (Initializable, AccessControlEnumerableUpgradeable, PausableUpgradeable, UUPSUpgradeable)
   - [Pub] initialize #
      - modifiers: initializer
   - [Ext] setVerifiedAddress #
      - modifiers: onlyRole
   - [Ext] updateTxFee #
      - modifiers: onlyRole
   - [Pub] setLandGovVotePercentage #
      - modifiers: onlyRole
   - [Pub] getTxFee
   - [Ext] getFinancingID
   - [Ext] getFinancingNFTID
   - [Pub] hasVoted
   - [Ext] requestFinancing #
      - modifiers: onlyRole
   - [Ext] voteFinancing #
   - [Ext] cancelFinancing #
      - modifiers: onlyRole
   - [Ext] closeFinancing #
      - modifiers: onlyRole
   - [Ext] stake #
   - [Pub] canUnstake
   - [Ext] unstake #
   - [Int] _unstake #
   - [Pub] distributeRewards #
      - modifiers: onlyRole
   - [Pub] withdrawTxFee #
      - modifiers: onlyRole
   - [Ext] withdrawErc20 #
      - modifiers: onlyRole
   - [Int] _authorizeUpgrade #
      - modifiers: onlyRole

+  LorcGeneralStaking (Initializable, AccessControlEnumerableUpgradeable, PausableUpgradeable, UUPSUpgradeable, DSMath
   - [Pub] initialize #
      - modifiers: initializer
   - [Ext] name
   - [Ext] setLorcAddress #
```

```
+  LandOrcNFT (Initializable, ERC721Upgradeable, ERC721EnumerableUpgradeable, ERC721URIStorageUpgradeable, PausableUpgradeable, AccessControlEnumerableUpgradeable, ERC721BurnableUpgradeable, UUPSUpgradeable)
    - [Pub] initialize #
      - modifiers: initializer
    - [Ext] safeMint #
      - modifiers: onlyRole
    - [Ext] updateTokenURI #
      - modifiers: onlyRole
    - [Ext] pause #
      - modifiers: onlyRole
    - [Ext] unpause #
      - modifiers: onlyRole
    - [Int] _beforeTokenTransfer #
      - modifiers: whenNotPaused
    - [Int] _burn #
    - [Pub] tokenURI
    - [Ext] setLorcAddress #
      - modifiers: onlyRole
    - [Int] _afterMint #
    - [Pub] supportsInterface
    - [Int] _authorizeUpgrade #
      - modifiers: onlyRole

+  LandOrc (Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlEnumerableUpgradeable)
    - [Pub] initialize #
      - modifiers: initializer
    - [Ext] setLandNFTAddress #
      - modifiers: onlyRole
    - [Ext] setRewardVaultAddress #
      - modifiers: onlyRole
    - [Ext] setExchangeAddress #
      - modifiers: onlyRole
    - [Ext] setRewardPercentage #
      - modifiers: onlyRole
    - [Ext] addBlackList #
      - modifiers: onlyRole
    - [Ext] removeBlackList #
```

```
+  LandGovLegal (Initializable, AccessControlEnumerableUpgradeable, PausableUpgradeable, UUPSUpgradeable)
    - [Pub] initialize #
      - modifiers: initializer
    - [Ext] setLorcFinancing #
    - [Ext] totalLegalTeam
    - [Pub] hasVoted
    - [Ext] getVoteCount
    - [Pub] getMinVerifierRequired
    - [Ext] verifyLorcFinancing #
      - modifiers: onlyRole,whenNotPaused
    - [Ext] withdrawErc20 #
      - modifiers: onlyRole
    - [Ext] pause #
      - modifiers: onlyRole
    - [Ext] unpause #
      - modifiers: onlyRole
    - [Int] _authorizeUpgrade #
      - modifiers: onlyRole

+  LandGov (Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlEnumerableUpgradeable, UUPSUpgradeable)
    - [Pub] initialize #
      - modifiers: initializer
    - [Ext] addBlackList #
      - modifiers: onlyRole
    - [Ext] removeBlackList #
      - modifiers: onlyRole
    - [Pub] pause #
      - modifiers: onlyRole
    - [Pub] unpause #
      - modifiers: onlyRole
    - [Ext] mint #
      - modifiers: onlyRole
    - [Int] _beforeTokenTransfer #
      - modifiers: whenNotPaused,notBlacklisted,notBlacklisted,notBlacklisted
    - [Int] _authorizeUpgrade #
```

## 7.0 Auditing Approach and Methodologies applied

Throughout the audit of the **LandOrc** smart contract, care was taken to ensure:

- Overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per intended behaviour mentioned in the whitepaper.
- Implementation of token standards.
- Efficient use of gas.
- Code is safe from Re-entrancy and other vulnerabilities.

A combination of manual and automated security testing was used to balance efficiency, timeliness, practicality as well as accuracy regarding the scope of the smart contract audit.While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of smart contracts and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

### 7.1 Structural Analysis

In this step we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure Smart contract is structured in a way that will not result in future problems.

### 7.2 Static Analysis

Static Analysis of smart contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test the security of smart contracts.

### 7.3 Code Review / Manual Analysis

Manual Analysis or review of the code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper.

### 7.4 Gas Consumption

In this step we have checked the behaviour of the smart contract in production. Checks were done to know how much gas gets consumed and possibilities of optimization of code to reduce gas consumption.

### 7.5 Tools and Platforms used for Audit

VSCode, Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Manticore, Slither.

## 7.6 Checked Vulnerabilities

We have scanned The People Reserves smart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC-20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

## 8.0 Limitations on Disclosure and Use of this Report

This report contains information concerning potential details of LandOrc and methods for exploiting them. Entersoft recommends that special precautions be taken to protect the confidentiality of both this document and the information contained herein. Security Assessment is an uncertain process, based on past experiences, currently available information, and known threats. All information security systems, which by their nature are dependent on human beings, are vulnerable to some degree. Therefore, while Entersoft considers the major security vulnerabilities of the analysed systems to have been identified, there can be no assurance that any exercise of this nature will identify all possible vulnerabilities or propose exhaustive and operationally viable recommendations to mitigate those exposures. In addition, the analysis set forth herein is based on the technologies and known threats as of the date of this report. As technologies and risks change over time, the vulnerabilities associated with the operation of the LandOrc Smart Contract described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities will also change. Entersoft makes no undertaking to supplement or update this report based on changed circumstances or facts of which Entersoft becomes aware after the date hereof, absent a specific written agreement to perform the supplemental or updated analysis. This report may recommend that Entersoft use certain software or hardware products manufactured or maintained by other vendors. Entersoft bases these recommendations upon its prior experience with the capabilities of those products. Nonetheless, Entersoft does not and cannot warrant that a particular product will work as advertised by the vendor, nor that it will operate in the manner intended. This report was prepared by Entersoft for the exclusive benefit of LandOrc and is proprietary information. The Non-Disclosure Agreement (NDA) in effect between Entersoft and LandOrc govern the disclosure of this report to all other parties including product vendors and suppliers.