



CERTIK

# Harvest by Kava Labs

## Security Assessment

October 16th, 2020

By :

Jay Jie@ CertiK

[jay.jie@certik.io](mailto:jay.jie@certik.io)

Georgios Delkos@ CertiK

[georgios.delkos@certik.io](mailto:georgios.delkos@certik.io)



## Disclaimer

CertiK reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

## What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has indeed completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.



# Overview

## Project Summary

|              |                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Project Name | <a href="#">Kava</a>                                                                                                                                                                                                                                                                                                    |
| Description  | Multi-chain DeFi lending platform                                                                                                                                                                                                                                                                                       |
| Platform     | Cosmos SDK v0.39.1                                                                                                                                                                                                                                                                                                      |
| Codebase     | <a href="#">kava</a>                                                                                                                                                                                                                                                                                                    |
| Commits      | 1. <a href="#">04946493ae927ee7c6dab027f444d32177d82540</a><br>2. <a href="#">0ded29678e4e168d20ad89292c470fcff244db95</a><br>3. <a href="#">e8a12b5b71ff9115ffe641418011cf63534456e8</a><br>4. <a href="#">b6e8759b86a1d75a12dde5a7fd022327f5701588</a><br>5. <a href="#">9bbdf1410802a148e61684ef6965a3b345acab2b</a> |

## Audit Summary

|                     |                                |
|---------------------|--------------------------------|
| Delivery Date       | Oct. 16, 2020                  |
| Audit Methods       | Static Analysis, Manual Review |
| Consultants Engaged | 2                              |
| Timeline            | Oct. 05, 2020 - Oct. 16 2020   |

## Vulnerability Summary

|                     |   |
|---------------------|---|
| Total Issues        | 4 |
| Total Critical      | 0 |
| Total Major         | 0 |
| Total Minor         | 0 |
| Total Informational | 4 |



# Executive Summary

## Preliminary:

Built on top of the Cosmos SDK, Kava is a multi-asset, interoperable Decentralized Finance (DeFi) platform offering collateralized loans and stablecoins (e.g. USDX), to end-users and other blockchains. The sole objective of the audit is to verify Kava Labs' implementation of the Harvest module, a cross-chain money market, against the provided specifications. A series of thorough security assessments were carried out, the goal of which is to help the said project protect their users by finding and fixing known vulnerabilities that could cause unauthorized access, loss of funds, cascading failures, and/or other vulnerabilities. Alongside each security finding, recommendations on fixes and best practices will also be given.

## Alleviations:

All recommendations were addressed and fully attended to in pull request [#686](#).



# Review Notes

The primary focus for the audit is to have a thorough look into the following parts of the application:

- Code Structure
- Application Module Interfaces
- Messages and Queries
- Invariants (if present)
- Keepers
- Module Interfaces
- Module Genesis
- Errors

Following a modular design approach outlined in the Cosmos SDK, we carefully inspect the module(s) within scope to ensure that:

1. Application module interfaces (`AppModuleBasic` and `AppModule` at least) are correctly implemented
2. Order of execution between key components of the module are properly managed by `Module Manager`
3. Messages are accompanied by constructor functions, have proper type definition, and correctly implement the `Msg` interface
4. Queries are accompanied by queriers, query commands and query return types
5. Handlers and their corresponding handler functions are properly added and implemented
6. Keepers appropriately expose getter/setter methods for the store(s) managed by the module
7. Invariants are properly implemented and registered
8. Module-specific errors are wrapped to provide additional specific execution context
9. The SDK is utilized in a least-authority manner, primarily for routing messages to their intended modules

Specifically in the Harvest module we analyze how the state machines are defined and how state transitions are triggered by messages, the goal of which is to check the implementation against the specs and hence minimize the possibilities of unintentional state behaviors taking place.



# State Transition Checks

## Claim

| Function                     | Check                                                                                         | Reference                         | Pass |
|------------------------------|-----------------------------------------------------------------------------------------------|-----------------------------------|------|
| <code>ClaimReward</code>     | get a claim from the store                                                                    | <a href="#">claim.go L25-L28</a>  | ✓    |
|                              | validate the claim owner and the reward receiver                                              | <a href="#">claim.go L30-L33</a>  | ✓    |
|                              | claim the rewards according to the <code>depositType</code> and send them to the reward owner | <a href="#">claim.go L35-L45</a>  | ✓    |
|                              | delete the claim from the store                                                               | <a href="#">claim.go L56</a>      | ✓    |
| <code>GetPeriodLength</code> | check if lockup period in multiplier is zero                                                  | <a href="#">claim.go L64-L66</a>  | ✓    |
|                              | calculate the length of the period based on block time and multiplier                         | <a href="#">claim.go L67-L78</a>  | ✓    |
| <code>claimLPReward</code>   | get the liquidity provider's distribution schedule and the multiplier for it                  | <a href="#">claim.go L83-L90</a>  | ✓    |
|                              | check if the claim had expired                                                                | <a href="#">claim.go L91-L93</a>  | ✓    |
|                              | calculate the reward amount and mint new coins for it                                         | <a href="#">claim.go L94-L98</a>  | ✓    |
|                              | get the length of reward period                                                               | <a href="#">claim.go L99-L102</a> | ✓    |
|                              | send the rewards to the recipient                                                             | <a href="#">claim.go L104</a>     | ✓    |



## State Transition Checks

### Claim

| Function                 | Check                                                                                                                                                                                                         | Reference                          | Pass |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------|
| {claimDelegatorReward}   | get the delegator's distribution schedule and the multiplier for it                                                                                                                                           | <a href="#">claim.go L108-L115</a> | ✓    |
|                          | check if the claim had expired                                                                                                                                                                                | <a href="#">claim.go L116-L118</a> | ✓    |
|                          | calculate the reward amount and mint new coins for it                                                                                                                                                         | <a href="#">claim.go L119-L123</a> | ✓    |
|                          | get the length of reward period                                                                                                                                                                               | <a href="#">claim.go L125-L128</a> | ✓    |
|                          | send the rewards to the recipient                                                                                                                                                                             | <a href="#">claim.go L130</a>      | ✓    |
| {validateSenderReceiver} | get the sender account from the store                                                                                                                                                                         | <a href="#">claim.go L134-L137</a> | ✓    |
|                          | if the sender account is a validator vesting account, check if the sender address matches to the receiver address. Otherwise, check if the two addresses don't match, in which case an error should be raised | <a href="#">claim.go L138-L147</a> | ✓    |



# State Transition Checks

## Deposit

| Function                       | Check                                                                                                                                                          | Reference                            | Pass |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|------|
| <code>ValidateDeposit</code>   | check if the <code>depositType</code> is liquidity provider (lp)                                                                                               | <a href="#">deposit.go L54-L62</a>   | ✓    |
| <code>ValidateLPDeposit</code> | check if the <code>depositDenom</code> matches the liquidity provider's denom                                                                                  | <a href="#">deposit.go L67-L75</a>   | ✓    |
| <code>Deposit</code>           | validate the deposit and transfer the coins from the depositor's account to the module account ONLY IF the <code>depositType</code> is liquidity provider (lp) | <a href="#">deposit.go L14-L27</a>   | ✓    |
|                                | check if there's existing deposits in the store under the depositor's account. If there is, add to it. If not, open a new deposit                              | <a href="#">deposit.go L29-L34</a>   | ✓    |
|                                | store the deposit                                                                                                                                              | <a href="#">deposit.go L36</a>       | ✓    |
| <code>Withdraw</code>          | get from the store the deposit to be withdrawn, and check if the requested withdrawal amount is larger than the actual amount in the store                     | <a href="#">deposit.go L79-L85</a>   | ✓    |
|                                | transfer the coins from the module account to the original depositor's account ONLY IF the <code>depositType</code> is liquidity provider (lp)                 | <a href="#">deposit.go L88-L96</a>   | ✓    |
|                                | subtract the withdrawn amount from the original value and update the deposit amount in the store                                                               | <a href="#">deposit.go L121-L122</a> | ✓    |



# State Transition Checks

## Rewards

| Function                                      | Check                                                                                                                                                                                                                                                                                                          | Reference                          | Pass |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------|
| <code>ApplyDepositRewards</code>              | get <code>previousBlockTime</code> and <code>Params</code> from the store; error out if not found and/or inactive                                                                                                                                                                                              | <a href="#">rewards.go L15-L24</a> | ✓    |
|                                               | range over <code>LiquidityProviderSchedules</code> (lps) and calculate <code>rewardsDistributed</code> . Skip if 1) the lps is inactive; 2) lps ends before (or starts after) <code>blockTime</code> ; 3) <code>totalDeposited</code> is zero; or 4) <code>rewardsToDistribute</code> is zero                  | <a href="#">rewards.go L27-L67</a> | ✓    |
|                                               | Update <code>previousBlockTime</code> in the store                                                                                                                                                                                                                                                             | <a href="#">rewards.go L68</a>     | ✓    |
| <code>ShouldDistributeValidatorRewards</code> | get <code>previousDelegatorDistribution</code> and <code>Params</code> from the store; error out if not found and/or inactive                                                                                                                                                                                  | <a href="#">rewards.go L73-L81</a> | ✓    |
|                                               | range over <code>DelegatorDistributionSchedules</code> (dds) and check if <code>timeElapsed</code> exceeds <code>DistributionFrequency</code> . Skip if 1) denom doesn't match the <code>DepositDenom</code> in dds; or 2) the <code>DistributionSchedule</code> in dds ends before the <code>blockTime</code> | <a href="#">rewards.go L82-L93</a> | ✓    |



# State Transition Checks

## Rewards

| Function                            | Check                                                                                                                                                                                                                              | Reference                            | Pass |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|------|
| <code>ApplyDelegationRewards</code> | get <code>delegatorSchedule</code> from the store;<br>error out if not found,<br><code>DistributionSchedule</code> is inactive, or<br><code>DistributionSchedule</code> starts after<br><code>blockTime</code>                     | <a href="#">rewards.go L99-L108</a>  | ✓    |
|                                     | get the coin amount in in the bonded pool<br>and check if it is zero                                                                                                                                                               | <a href="#">rewards.go L109-L113</a> | ✓    |
|                                     | get <code>previousDelegatorDistribution</code><br>from the store and return if not found                                                                                                                                           | <a href="#">rewards.go L114-L117</a> | ✓    |
|                                     | calculate <code>rewardToDistribute</code>                                                                                                                                                                                          | <a href="#">rewards.go L119</a>      | ✓    |
|                                     | iterate over all validators and store the key-<br>value pair <code>ValAddress→conversion</code><br>factor in a map ( <code>sharesToTokens</code> );<br>Continue iteration when the validator has<br>zero tokens and/or is unbonded | <a href="#">rewards.go L123-L134</a> | ✓    |
|                                     | iterate over all delegations; calculate and<br>add <code>rewardsEarned</code> to the claim if it is<br>not zero                                                                                                                    | <a href="#">rewards.go L138-L153</a> | ✓    |



# State Transition Checks

## Timelock

| Function                                                 | Check                                                                                                                                                                                                                                                                                                                                                                           | Reference                             | Pass |
|----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------|
| <code>addCoinsToVestingSchedule</code>                   | get <code>PeriodicVestingAccount</code> from the store                                                                                                                                                                                                                                                                                                                          | <a href="#">timelock.go L77-L78</a>   | ✓    |
|                                                          | add the new vesting coins to <code>OriginalVesting</code>                                                                                                                                                                                                                                                                                                                       | <a href="#">timelock.go L80</a>       | ✓    |
|                                                          | 1) if all vesting periods under the vesting account have completed before <code>blockTime</code> , append a new period to the vesting account → update <code>EndTime</code> → update the account in the store; 2) if the earliest vesting period under the vesting account starts after <code>blockTime</code> , update all vesting periods → set <code>StartTime</code> to now | <a href="#">timelock.go L81-L104</a>  | ✓    |
|                                                          | insert a new vesting period into the existing vesting schedule                                                                                                                                                                                                                                                                                                                  | <a href="#">timelock.go L107-L143</a> | ✓    |
| <code>SendTimeLockedCoinsToPeriodicVestingAccount</code> | send time-locked coins from the module account to the recipient                                                                                                                                                                                                                                                                                                                 | <a href="#">timelock.go L47-L50</a>   | ✓    |
|                                                          | add coins to the input account's vesting schedule                                                                                                                                                                                                                                                                                                                               | <a href="#">timelock.go L51</a>       | ✓    |
| <code>SendTimeLockedCoinsToBaseAccount</code>            | send time-locked coins from the module account to the recipient                                                                                                                                                                                                                                                                                                                 | <a href="#">timelock.go L57-L60</a>   | ✓    |
|                                                          | transition the account to a periodic vesting account and update it in the store                                                                                                                                                                                                                                                                                                 | <a href="#">timelock.go L63-L70</a>   | ✓    |
| <code>SendTimeLockedCoinsToAccount</code>                | get the senderModule account from the store and check if it has sufficient balance                                                                                                                                                                                                                                                                                              | <a href="#">timelock.go L19-L22</a>   | ✓    |
|                                                          | get the recipient account from the store; check if the recipient account is a valid account and if the input length is greater than zero; send time-locked coins to the recipient account according to its account type                                                                                                                                                         | <a href="#">timelock.go L25-L42</a>   | ✓    |



# Findings

## Status Icon Definitions

|                                                                                                |                                                                                               |                                                                                                   |
|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
|  Resolved     |  In Progress |  Ignored (pro) |
|  Not Resolved |  Incorrect   |  Ignored (con) |

## Findings Overview

| ID                     | Title                             | Type                    | Severity      | Status                                                                               |
|------------------------|-----------------------------------|-------------------------|---------------|--------------------------------------------------------------------------------------|
| <a href="#">KAV-01</a> | Verbose Code                      | Language Best Practices | Informational |   |
| <a href="#">KAV-02</a> | Inefficient Conditional Statement | Language Best Practices | Informational |   |
| <a href="#">KAV-03</a> | Inefficient Conditional Statement | Language Best Practices | Informational |   |
| <a href="#">KAV-04</a> | Unused Parameter                  | Language Best Practices | Informational |  |



## KAV-01: Verbose Code

| Type                    | Severity      | Location                       |
|-------------------------|---------------|--------------------------------|
| Language Best Practices | Informational | <a href="#">querier.go L81</a> |

### Description:

In the following snippet, the declaration and assignment for a variable (e.g. `depositDenom`) before/after a given logical statement can be replaced with a one-liner.

```
depositDenom := false
owner := false
depositType := false
if len(params.DepositDenom) > 0 {
    depositDenom = true
}
if len(params.Owner) > 0 {
    owner = true
}
if len(params.DepositType) > 0 {
    depositType = true
}
```

### Recommendation:

Replace with the following for better readability.

```
depositDenom := len(params.DepositDenom) > 0
owner := len(params.Owner) > 0
depositType := len(params).DepositType > 0
```

### Alleviation:

The recommendation was applied in commit [0ded29678e4e168d20ad89292c470fcff244db95](#).



## KAV-02: Inefficient Conditional Statement

| Type                    | Severity      | Location                       |
|-------------------------|---------------|--------------------------------|
| Language Best Practices | Informational | <a href="#">querier.go L97</a> |

### Description:

The following snippet exhibits a fair amount of complexity with the nested if/else statements, which sacrifices code readability.

```
if depositDenom && owner && depositType {  
...  
} else if depositDenom && owner {  
...  
}  
...
```

### Recommendation:

Replace the nested if/else statements with a switch statement.

```
switch {  
case depositDenom && owner && depositType:  
...  
case depositDenom && owner:  
...  
case depositDenom && depositType:  
...  
default:  
...  
}
```

### Alleviation:

The recommendation was applied in commit [e8a12b5b71ff9115ffe641418011cf63534456e8](#).



## KAV-03: Inefficient Conditional Statement

| Type                    | Severity      | Location                        |
|-------------------------|---------------|---------------------------------|
| Language Best Practices | Informational | <a href="#">querier.go L249</a> |

### Description:

The following snippet exhibits a fair amount of complexity with the nested if/else statements, which sacrifices code readability.

```
if depositDenom && owner && depositType {  
...  
} else if depositDenom && owner {  
...  
}  
...
```

### Recommendation:

Replace the nested if/else statements with a switch statement.

```
switch {  
case depositDenom && owner && depositType:  
...  
case depositDenom && owner:  
...  
case depositDenom && depositType:  
...  
default:  
...  
}
```

### Alleviation:

The recommendation was applied in commit [b6e8759b86a1d75a12dde5a7fd022327f5701588](#).



## KAV-04: Unused Parameter

| Type                    | Severity      | Location                       |
|-------------------------|---------------|--------------------------------|
| Language Best Practices | Informational | <a href="#">querier.go L33</a> |

### Description:

The following snippet parameter `req` is passed in function `queryGetParams` but not used.

```
func queryGetParams(ctx sdk.Context, req abci.RequestQuery, k Keeper) ([]byte, error) {  
    // Get params  
    params := k.GetParams(ctx)  
    // Encode results  
    bz, err := codec.MarshalJSONIndent(k.cdc, params)  
    if err != nil {  
        return nil, sdkerrors.Wrap(sdkerrors.ErrJSONMarshal, err.Error())  
    }  
    return bz, nil  
}
```

### Recommendation:

Remove parameter `req`.

### Alleviation:

The recommendation was applied in commit [9bbdf1410802a148e61684ef6965a3b345acab2b](#).