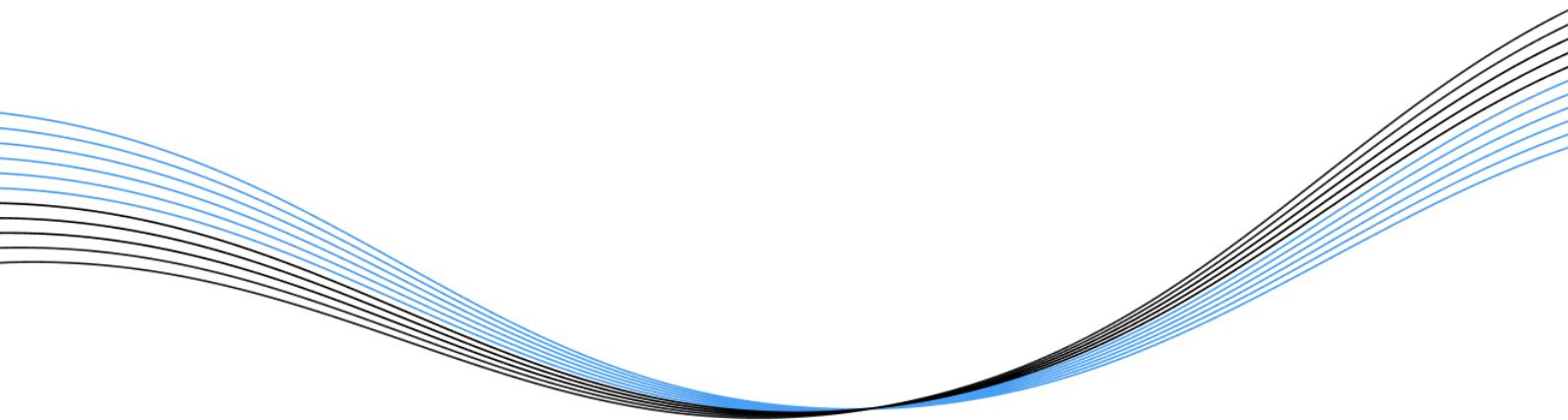


Hexlant.

© Hexlant Inc.
340 Gangnam-daero
Seoul, Republic of Korea
06242
Hexlant.com

KLAYBANK-STAKE SMART CONTRACT AUDIT REPORT



Audit Date
7 Feb 2022

Category

De-fi

Auditor

Hexlant Audit Team

This audit report specifies that the Hexlant Technical Team validated and notified that it has no technical defects.

AUDIT

OVERVIEW

PUBLISHED INFORMATION

REPORT NUMBER	HEXLANT20220124_01
DATE	2022/02/7
AUDIT SCOPE	KlaybankDistributionManager.sol KlaybankEcosystemReserve.sol KlaybankIncentivesController.sol StakedToken.sol StakedKbt.sol

PROJECT INFORMATION

TITLE	KlayBank stake
TYPE	DEFI
PLATFORM	KLAYTN
CONTRACT ADDRESS	
REPOSITORY	https://github.com/klaybank/klaybank-audit-hexlant
AUDIT COMMIT HASH	03e9af270ca631a124a8fde52a268623b451c29f f9a613b649266d868f1cef7892281f18bd5054fc 66fb4fcaca941cc65f7fbf5b30f6e34679f62edd LATEST 67cfeafe687c5c143154afd81caa7efaeaaa30fa

VULNERABILITY ANALYSIS

CRITICAL (Resolve)	2 → 0	unregisterShareSnapshot() 로직 에러 외 1
HIGH	0	-
MEDIUM (Resolve)	1 → 0	shareRatio 예외 처리
LOW	5→ 1	registerShareSnapshot() event 누락 외 4

CENTRALIZED FUNCTION

configureAsset(s)	Function to configure emitting reward(s) for assets to stake
setStakedTokenRewardRatio	Function to set _stakedTokenRewardRatio, the ratio of staking from claimed rewards
setKlaybankGovernance	Function to set klaybankGovernance
setDistributionStartTimestamp	Function to set timestamp to start emitting rewards for asset
setClaimer	Function to set claimer for specific user

COMPANY PROPOSAL

Hexlant는 2018년에 설립한 블록체인 기술 기업입니다. 삼성전자 출신의 보안·네트워크·소프트웨어 전문가가 스마트 컨트랙트와 블록체인 프로토콜의 보안 결함을 발견하고 블록체인 생태계의 기술 안정성을 입증하기 위해 설립하였습니다.

Hexlant는 블록체인 동작 환경을 파악하기 위해 20개 이상의 블록체인 메인넷을 직접 구축하고 있습니다. 나아가 키 보안 알고리즘 및 메인넷 모니터링 기술을 개발했습니다. 이 방식은 비트코인, 이더리움, 폴카닷, 에이다 등 헥슬란트가 보유한 모든 메인넷 플랫폼에서 적용되고 운영됩니다.

Hexlant는 위와 같은 기술 운영 경험을 바탕으로 스마트 컨트랙트 기술을 검증합니다. 스마트 컨트랙트 내 버그를 발견하는 오류 테스트 뿐만 아니라 메인넷 상황에서의 문제점을 탐지하며 서비스 관점에서 지속적으로 운영할 수 있는 블록체인 기술 가이드를 제공합니다.

Hexlant의 고객사는 컨트랙트에 대한 취약성 감사부터 오너 키 관리, 블록체인 지갑 시스템 구축 등 블록체인 기술 전반의 서비스를 제공받을 수 있습니다. 현재 200여개의 고객사가 Hexlant의 서비스를 바탕으로 블록체인 사업을 시작, 운영했으며 누적으로 관리하는 자산은 12조를 달성했습니다.

Initials for identification purposes:



for identification purposes
Hexlant.

CONTENTS

- 1. Analysis Purpose**
- 2. Vulnerability Classification**
- 3. Test Result**
- 4. Test Case**
- 5. Vulnerability Analysis**
- 6. Conclusion**

ANALYSIS PURPOSE

본 리포트는 발행된 컨트랙트 코드가 요구사항을 충분히 만족하는지, 그리고 보안의 취약점과 실제 운영하면서 발생 할 수 있는 문제들을 파악하고 해결방안을 찾기위해 분석을 수행하고 그 결과를 정리하였습니다. 이번 코드 분석은 다음과 같은 요소들을 검증하기위해 진행하였습니다.

- 구현된 기능의 정상 작동 여부
- 기능 수행 중 보안 위험성
- Off Chain에서 발생하는 문제에 대한 대비
- 컨트랙트 코드의 가독성 및 코드 완성도

VULNERABILITY CLASSIFICATION

본 취약성 검증은 오류 위험도를 아래와 같이 분류, 평가합니다.

• Critical Severity

심각성 치명적 단계는 큰 보안 결함을 뜻하며 자산 탈취 및 동결, 추가 발행 등 치명적인 문제를 야기합니다. 본 결함은 반드시 수정되야 합니다.

• High Severity

심각성 높은 단계는 특수 조건에 의해 보안 결함이 발생할 수 있는 항목이며 수정을 강력하게 권고합니다.

• Medium Severity

심각성 중간 단계는 보안 결함은 아니나 비효율적인 컨트랙트 동작을 야기합니다. 컨트랙트를 효율적으로 동작하도록 수정을 권유하는 항목입니다.

• Low Severity

심각성 낮음 단계는 보안에는 문제가 없으나 컨트랙트 구조 개선을 위해 수정을 권유하는 항목입니다.

Severity	Issue	Status (Found)	Status (Resolve)
● CRITICAL (2)	unregisterShareSnapshot() 로직 에러	(Found - 03e9af2)	(Resolve - f9a613b)
	_getAssetIndex() 로직 에러	(Found - 03e9af2)	(Resolve - 66fb4fc)
● HIGH (0)	-		
● MEDIUM (1)	shareRatio 예외 처리	(Found - 03e9af2)	(Resolve - f9a613b)
● LOW (5)	shareSnapshot 변경 시 event 누락	(Found - 03e9af2)	(Resolve - 67cfeaf)
	registerShareSnapshot() 함수 사용하지 않는 매개변수	(Found - f9a613b)	(Resolve - 66fb4fc)
	_getAssetIndex() 사용하지 않는 매개변수	(Found - 03e9af2)	(Resolve - f9a613b)
	getNextHeatupTimestamp() 변수명 중복	(Found - 03e9af2)	

handleAction() 인터페이스 변수명 불일치

(Found - 03e9af2)

(Resolve - 67cfeaf)

TEST RESULT

Code Coverage

코드 커버리지는 작성한 테스트가 얼마만큼 컨트랙트 코드의 기능을 테스트 했는지 알 수 있는 정량적인 지표입니다.

라이브러리와 일부 컨트랙트에 구현된 기능에 대해 추가적인 호출이 진행되지 않은 경우가 존재합니다.

아래의 Coverage 지표는 위 사항을 반영한 결과입니다.

File Name	Statements	Functions	Lines
KlaybankDistributionManger.sol	97.91%	100%	97.22%
KlaybankEcosystemReserve.sol	97.4%	100%	95.73%
KlaybankIncentivesController.sol	96.08%	100%	96.15%
StakedKbt.sol	100%	100%	100%
StakedToken.sol	98.82%	100%	98.82%

TEST CASE

실제 적용한 테스트케이스 목록입니다. 총 45개 테스트 시나리오를 적용하여 검수했습니다.

Test Case 1. KlaybankIncentivesController.sol (13개의 테스트)

INITIALIZE

1.1. initialize	Result
initializer가 한번만 호출되는가	PASS FAIL

CONFIGURE ASSETS

2.1. configureAssets	Result
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS FAIL
올바르지 않은 설정 매개변수 입력시 예외처리 되는가	PASS FAIL
이벤트를 발생시키는가	PASS FAIL

HANDLEACTION

3.1. handleAction	Result
assetData를 정확하게 업데이트 하는가	PASS FAIL
이벤트를 발생시키는가	PASS FAIL

CLAIM REWARDS

4.1. claimRewards	Result
0x0 주소로 리워드를 보낼 시 예외처리 되는가	PASS FAIL
유저가 리워드를 지급하도록 등록된 토큰 보유 시 reward 토큰을 정확하게 전달하는가	PASS FAIL
이벤트를 발생시키는가	PASS FAIL

4.2. claimRewardsOnBehalf	Result
0x0 주소로 리워드를 보낼 시 예외처리 되는가	PASS FAIL
0x0 주소의 리워드를 요청할 시 예외처리 되는가	PASS FAIL
유저에게 reward 토큰을 정확하게 전달하는가	PASS FAIL

authorized되지 않은 주소로부터 호출 시 예외처리 되는가	PASS	FAIL
-------------------------------------	------	------

Test Case 2. StakedKBT.sol (55개의 테스트)

INITIALIZE

5.1. initialize	Result	
0x0 주소를 입력 받을 시 예외처리 되는가	PASS	FAIL
initializer가 한번만 호출되는가	PASS	FAIL

CONFIGUREASSET

6.1. configureAsset	Result	
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL

6.2. KlaybankDistributionManager._configureAsset	Result	
AssetData를 정확하게 업데이트 하는가	PASS	FAIL
유효하지 않은 shareRatio 입력시 예외처리 되는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

6.3. KlaybankDistributionManager.setDistributionStartTimestamp	Result	
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL
유효하지 않은 startTimestamp을 입력받을 시 예외처리 되는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

SHARESNAPSHOT

7.1 KlaybankDistributionManager.registerShareSnapshot	Result	
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL
유효하지 않은 매개변수를 입력받을 시 예외처리 되는가	PASS	FAIL
asset에 새로운 shareSnapshot을 추가하는가	PASS	FAIL

7.2 KlaybankDistributionManager.unregisterShareSnapshot	Result	
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL
제거하고자 하는 shareSnapshot을 제거하는가	PASS	FAIL

7.3 shareSnapshots getter	Result	
getCurrentShareRatio(), getShareSnapshots(),	PASS	FAIL

getSortedShareSnapshots()가 정확한 값을 반환하는가

distributeStartTime 변경 시 getSortedShareSnapshots()이 정확한 값을 반환하는가	PASS	FAIL
---	------	------

STAKING

8.1. stake	Result	
0x0 주소를 입력 받을 시 예외처리 되는가	PASS	FAIL
0을 입력받을 시 예외처리 되는가	PASS	FAIL
GOVERNANCE_TOKEN을 정확히 전달하는가	PASS	FAIL
staking 시 유저에게 토큰을 정확한 수량만큼 발행하는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

CLAIMREWARDS

9.1. claimRewards	Result	
0x0 주소를 입력 받을 시 예외처리 되는가	PASS	FAIL
현재 누적된 보상보다 많은 수량을 출금 시, 누적된 모든 보상을 출금하는가	PASS	FAIL
claim시 보상의 일부분을 stakeToken 컨트랙트로 다시 staking 하는가	PASS	FAIL
LastUpdateTimestamp == distributionStartTimestamp 일 때, lastUpdateTime으로부터 한달 이내에 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
LastUpdateTimestamp == distributionStartTimestamp 일 때, lastUpdateTime으로부터 한달 이후 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
LastUpdateTimestamp == distributionStartTimestamp 일 때, distributionEndTimestamp 이후 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
LastUpdateTimestamp > distributionStartTimestamp 일 때, lastUpdateTime 이후 한달 이내에 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
LastUpdateTimestamp > distributionStartTimestamp 일 때, lastUpdateTime으로부터 한달 이후 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
LastUpdateTimestamp > distributionStartTimestamp 일 때, distributionEndTimestamp 이후 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
emissionPerSecond, shareSnapshot이 설정되어 있는 경우, 보상 출금 시 정확한 수량의 보상토큰을 전달하는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

REDEEM

10.1. redeem	Result	
0x0 주소를 입력 받을 시 예외처리 되는가	PASS	FAIL
0을 입력받을 시 예외처리 되는가	PASS	FAIL
heatup 시간이 지나지 않고 redeem 시 예외처리 되는가	PASS	FAIL
redeem 시 정확한 수량의 GOVERNANCE_TOKEN을 전달하는가	PASS	FAIL
스테이킹한 수량보다 많은 수량을 되찾을 시, 스테이킹한 수량 만큼의 GOVERNANCE_TOKEN만을 전달하는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

MIGRATION

11.1. ApproveMigrationTo	Result	
emissionManager가 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL
11.2. migrate	Result	
자기자신의 주소로 migrate시 예외처리 되는가	PASS	FAIL
stakeToken이 자기자신의 주소로 설정되어있지 않을 때 예외처리 되는가	PASS	FAIL
migrate 수량이 0일 시 예외처리 되는가	PASS	FAIL
호출한 유저의 토큰을 소각하고, migrate 하는 토큰을 정확한 수량 발행하는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

11.3. onMigration	Result	
호출 주소가 stakedToken이 아닐 경우 예외처리 되는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL

ERC20

12.1. transfer	Result	
0x0 주소로 토큰 전송 시 예외처리 되는가	PASS	FAIL
이벤트를 발생시키는가	PASS	FAIL
토큰 전송 시 heatup이 정확하게 업데이트 되는가	PASS	FAIL

모든 토큰 전송 시, heatup을 0으로 설정하는가	PASS	FAIL
klaybankGovernance존재 시 klaybankGovernance.onTransfer() 함수를 호출하는가	PASS	FAIL

12.2. permit	Result	
0x0 주소로 토큰 전송 시 예외처리 되는가	PASS	FAIL
deadline이 지난 후 permit 호출 시 예외처리 되는가	PASS	FAIL
permit 호출시 기존 permit을 취소하는가	PASS	FAIL
서명 데이터와 다른 데이터로 호출 시 예외처리 되는가	PASS	FAIL
유효하지 않은 nonce 값과 호출 시 예외처리 되는가	PASS	FAIL

DELEGATION

13.1. governance power	Result	
0x0 주소로 위임 시 예외처리 되는가	PASS	FAIL
스테이킹 후 governance power를 정확히 업데이트 하는가	PASS	FAIL
transfer 후 governance power를 정확히 업데이트 하는가	PASS	FAIL
redeem 후 governance power를 정확히 업데이트 하는가	PASS	FAIL
governance power 변동 시 이벤트를 발생시키는가	PASS	FAIL

13.2. delegateBySig, delegateByTypeBySig	Result	
deadline이 지난 후 delegateBySig 호출 시 예외처리 되는가	PASS	FAIL
유효하지 않은 nonce 값과 호출 시 예외처리 되는가	PASS	FAIL

Test Case 3. KlaybankEcosystemReserve.sol (7개의 테스트)

INITIALIZE

14.1. initialize	Result	
initializer가 한번만 호출되는가	PASS	FAIL

TOKEN

15.1. approve	Result	
fundsAdmin이 아닌 주소에서 호출 시 예외처리 되는가	PASS	FAIL
token을 recipient에게 amount만큼 승인하는가	PASS	FAIL

15.2. transfer	Result
fundsAdmin이 아닌 주소에서 흐출 시 예외처리 되는가	PASS FAIL
token을 recipient에게 amount만큼 전달하는가	PASS FAIL

SETFUNDSADMIN

16.1. setFundsAdmin	Result
fundsAdmin이 아닌 주소에서 흐출 시 예외처리 되는가	PASS FAIL
fundsAdmin을 변경하는가	PASS FAIL

VULNERABILITY ANALYSIS

총 8개 항목의 취약점을 발견하였으며, 발견된 8개의 취약 사항은 아래와 같이 모두 해결되었습니다.

- KlaybankDistributionManger.sol - 01 : `unregisterShareSnapshot()` 함수 로직 에러
- Status : Found - 03e9af2 → [Resolve - f9a613b](#)

Type	Severity	Location
Logic Error	• Critical Severity (Resolve)	KlaybankDistributionManager.sol L107
· Description	<p><code>unregisterShareSnapshot()</code> 함수는 asset에 등록된 shareSnapshot 중 하나를 지정하여 제거하는 함수입니다. <code>shareSnapshot</code>은 리스트에 저장되어 있고, <code>ShareSnapshot.at</code>의 크기에 따라 정렬되어 있습니다.</p> <p>L108에서 삭제하도록 지정한 <code>shareSnapshot</code>의 데이터만 변경이 이루어지며 올바른 삭제와 <code>shareSnapshot</code> 리스트의 정렬이 이루어지지 않습니다.</p>	
· Recommendation	<p>KlaybankDistributionManger.sol L107 <code>unregisterShareSnapshot()</code></p> <pre>for (uint256 index = 0; index < assetData.shareSnapshots.length - 1; index++) { if (assetData.shareSnapshots[index].at == at) { has = true; } if (has) { assetData.shareSnapshots[index] = assetData.shareSnapshots[index + 1]; } } if (has) { assetData.shareSnapshots.pop(); }</pre>	

삭제하고자 하는 `snapshot`의 `at`과 동일한 `at`을 가진 `snapshot`을 `assetData.shareSnapshots`에서 찾았을 때, 해당 인덱스 이후의 모든

shareSnapshot의 데이터를 다음 인덱스의 shareSnapshot 데이터로 변경하도록
수정

· Resolved

KlaybankDistributionManger.sol L107 unregisterShareSnapshot()

```
function unregisterShareSnapshot(address asset, uint256 at) external
onlyEmissionManager {
    AssetData storage assetData = assets[asset];
    uint256 length = assetData.shareSnapshots.length;
    if (length == 0) {
        return;
    }

    bool has = false;
    for(uint256 index = 0; index < length - 1; index++) {
        if (assetData.shareSnapshots[index].at == at) {
            has = true;
        }
        if (has) {
            assetData.shareSnapshots[index] = assetData.shareSnapshots[index + 1];
        }
    }
    if (has) {
        assetData.shareSnapshots.pop();
        return;
    }

    // when length is 1 or latest has same 'at'
    if (length > 0 && assetData.shareSnapshots[length - 1].at == at) {
        assetData.shareSnapshots.pop();
    }
}
```

unregisterShareSnapshot() 함수 로직 수정

- **KlaybankDistributionManger.sol - 02 : _getAssetIndex() function logic error**
- **Status :** Found - 03e9af2 → [Resolve - 66fb4fc](#)

Type	Severity	Location
Logic Error	• Critical Severity (Resolve)	KlaybankDistributionManager.sol L551
· Description	<p>_getAssetIndex() 함수는 distributionStartTimestamp로부터 현재까지 asset에 누적된 reward를 계산해주는 함수입니다. L.545부터 for문을 통해 lastupdateTimestamp로부터 업데이트되지 않은 달의 reward를 계산합니다.</p> <p>_calculateIndexDelta() 함수는 일정 기간동안 누적된 reward를 계산하는 함수로 세번째 매개변수는 시작시간, 네번째 매개변수는 종료시간입니다. L.550에서 계산하고자 하는 달의 시작시간과 종료시간이 정확히 설정되지 않았습니다.</p>	
· Recommendation	<p>KlaybankDistributionManger.sol L546 _getAssetIndex()</p> <pre>vars.indexDelta = vars.indexDelta.add(_calculateIndexDelta(asset, monthlyEmissionPerSecond[i], distributionStartTimestamp.add(SECONDS_OF_ONE_MONTH.mul(i)), distributionStartTimestamp.add(SECONDS_OF_ONE_MONTH.mul(i + 1)), totalBalance));</pre>	

L.550, L551 수정

- [Resolved](#)

KlaybankDistributionManger.sol L558 _getAssetIndex()

```
vars.indexDelta = vars.indexDelta.add(
    _calculateIndexDelta(
        asset,
        monthlyEmissionPerSecond[i],
        distributionStartTimestamp.add(SECONDS_OF_ONE_MONTH.mul(i)),
        distributionStartTimestamp.add(SECONDS_OF_ONE_MONTH.mul(i + 1)),
        totalBalance
    )
);
```

➤ **KlaybankDistributionManger.sol - 03 : shareRatio 예외 처리**

➤ **Status :** Found - 03e9af2 → **Resolve - f9a613b**

Type	Severity	Location
Validation of Value	● MEDIUM (Resolve)	KlaybankDistributionManager.sol L96 KlaybankDistributionManager.sol L315
· Description	shareRatio는 각 자산에 분배되는 reward의 비율로 percentage값입니다. 따라서 shareRatio는 100%보다 작거나 같은 값을 가져야 합니다. <code>registerShareSnapshot()</code> 함수와 <code>_configureAssets()</code> 함수에서 shareRatio를 새로 설정합니다. 두 함수에서 새로 설정하려는 값이 아닌, 이미 설정된 shareRatio가 PERCENTAGE_FACTOR(= 10000) 이하인지 비교하므로, 새로 설정하려는 shareRatio의 범위를 검사하도록 변경해야 합니다.	
· Recommendation	L.96 <div style="border: 1px solid black; padding: 5px;"><code>require(shareRatio <= PERCENTAGE_FACTOR, 'INVALID_SHARE_RATIO');</code></div> L.315 <div style="border: 1px solid black; padding: 5px;"><code>require(assetConfigInput.shareRatio <= PERCENTAGE_FACTOR, 'INVALID_SHARE_RATIO');</code></div>	
· Resolved	L.96, L.315 수정	

- **KlaybankDistributionManger.sol - 04 : shareSnapshot** 변경 시 **event** 누락
- **Status :** Found - 03e9af2 → **Resolve - 67cfea**

Type	Severity	Location
Event Omission	● LOW	KlaybankDistributionManager.sol L89 KlaybankDistributionManager.sol L100
· Description	<code>registerShareSnapshot()</code> , <code>unregisterShareSnapshot()</code> 함수를 사용하여 <code>shareSnapshot</code> 변경 시 로그를 남길 수 있도록 이벤트를 추가하는 것을 추천합니다.	
· Recommendation	RegisterShareSnapshot, UnregisterShareSnapshot 이벤트 추가	
· Resolved	SnapshotRegistered, SnapshotUnregistered 이벤트 추가 <code>registerShareSnapshot()</code> , <code>unregisterShareSnapshot()</code> 함수 호출로 <code>shareSnapshot</code> 변경 시 이벤트 호출	

- **KlaybankDistributionManger.sol - 05 : registerShareSnapshot()** 함수 사용하지 않는 매개변수
- **Status :** Found - f9a613b → **Resolve - 66fb4fc**

Type	Severity	Location
Unused Variable	● LOW (Resolve)	KlaybankDistributionManager.sol L93
· Description	<code>registerShareSnapshot()</code> 함수의 지역변수인 <code>currentShareRatio</code> 가 사용되지 않습니다.	
· Recommendation	사용하지 않는 지역변수 <code>currentShareRatio</code> 삭제	
· Resolved	L93 사용하지 않는 지역변수 <code>currentShareRatio</code> 삭제	

- **KlaybankDistributionManger.sol - 06 : _getAssetIndex()** 함수 사용하지 않는 매개변수
- **Status :** Found - 03e9af2 → **Resolve - f9a613b**

Type	Severity	Location
Unused Variable	● LOW (Resolve)	KlaybankDistributionManager.sol L483
· Description	<code>_getAssetIndex()</code> 함수의 매개변수인 <code>shareRatio</code> 가 함수 내에서 사용되지 않습니다.	
· Recommendation	사용하지 않는 매개변수 삭제 또는 comment 추가	
· Resolved	L.486 <code>_getAssetIndex()</code> 함수의 <code>shareRatio</code> 매개변수 삭제	

➤ **StakedToken.sol - 01 : getNextHeatupTimestamp()** 함수 변수명 충복

➤ **Status :** Found - 03e9af2

Type	Severity	Location
duplicate variable name	● LOW	StakedToken.sol L378
· Description :	<p><i>getNextHeatupTimestamp()</i> 함수의 매개변수인 <code>fromHeatupTimestamp</code>와 함수 지역 변수인 <code>fromHeatupTimestamp</code>의 이름이 같습니다. 실제 컨트랙트 작동에는 영향을 미치지 않습니다.</p>	
· Recommendation :	<p>함수의 매개변수 또는 지역변수의 이름을 변경</p>	

➤ **KlaybankIncentivesController.sol - 01 : handleAction()** 함수 인터페이스 변수명 불일치

➤ **Status :** Found - 03e9af2 → [Resolve - 67cfea](#)

Type	Severity	Location
Validation of parameter	● LOW	KlaybankIncentivesController.sol L76 IKlaybankIncentivesController.sol L51
· Description	<p><i>handleAction()</i> 함수의 인터페이스 매개변수명과 실제 구현된 함수의 매개변수명에 차이가 있습니다. 컨트랙트 기능에는 문제가 없지만, 외부 서비스에서 KlaybankIncentivesController 컨트랙트 사용 시 착오를 불러올 수 있습니다.</p>	
· Recommendation	<p>IKlaybankIncentivesController.sol L51</p>	

```
function handleAction (
    address user,
    uint256 totalSupply,
    uint256 userBalance
) external;
```

L53. 54 구현 함수와 동일한 변수명으로 수정

· [Resolved](#)

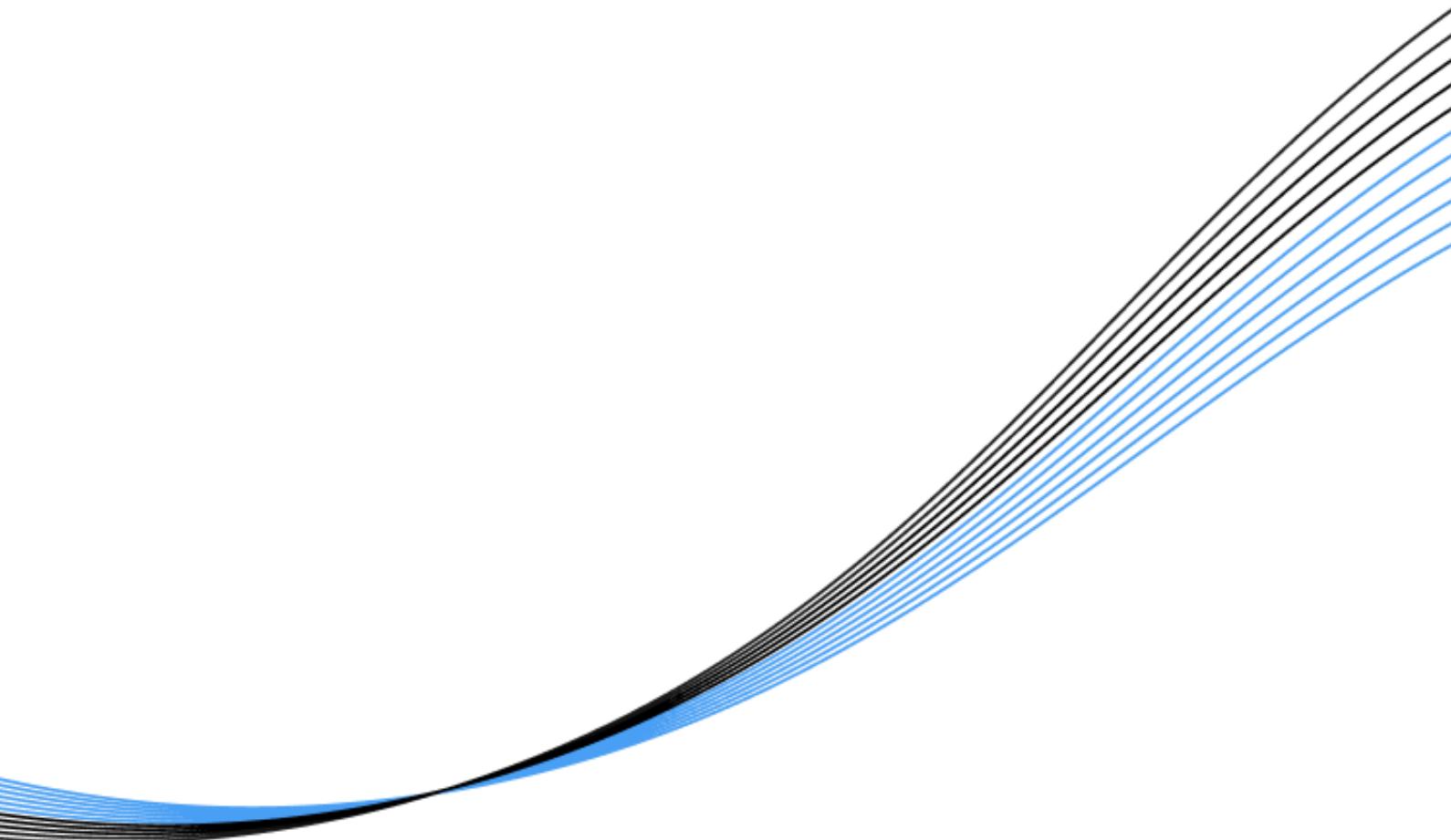
[IKlaybankIncentivesController.sol L51](#)

```
function handleAction(
    address asset,
    uint256 totalSupply,
    uint256 userBalance
) external;
```

인터페이스에 정의된 *handleAction()* 함수 매개변수명 변경

Declare

해당 리포트는 Hexlant의 스마트 컨트랙트 보안 감사 결과를 바탕으로 작성되었습니다. 해당 리포트는 비즈니스 모델의 적합성과 법적 규제, 투자에 대한 의견을 보증하지 않습니다. 리포트에 기술한 문제점 이외에 메인넷 기술 또는 가상머신을 비롯하여 발견되지 않은 문제점이 있을 수 있습니다. 해당 리포트는 논의 목적으로만 사용됩니다.



Hexlant.

-
contact@hexlant.com
www.hexlant.com