Tokamak Network - TOSv2

Security Assessment

2022. 09. 29

Carl Park @4000d

Disclaimer

이 보고서는 특정 팀 혹은 프로젝트의 투자, 비즈니스 모델의 적합성을 보장하는 것으로 간주 되어선 안됩니다.

이 보고서는 투자 혹은 특정 프로젝트에 참여를 위한 의사결정을 위해 사용되어서는 안됩니다.

이 보고서는 개발 과정에서 간과한 취약점과 추가 보안 조치가 필요한 영역을 발견할 수 있습니다. 하지만 이는 모든 취약점을 발견하는 것을 의미하지 않으며, 취약점이 발견되지 않더라도 소스 코드가 완벽히 안전하다는 것을 보장하지 않습니다. 코드 감사가 완료된 이후라도 존재하는 모든 취약점과 공격 유형에 대한 완벽한 보증이 아닙니다.

Overview

Github Repository: https://github.com/Onther-Tech/tosv2-contracts

Branch	Commit Hash
audit/v0.0.1	6a64bf34a35d41c64ab0d4941e2d8881ff632708
audit/v0.0.1_feedback_carl	19bb9828b345e8f85935540e957dd9774258e15f

Project Summary

해당 프로젝트에서 사용하는 토큰 목록과 주요 컨트랙트 기능은 다음과 같습니다.

Tokens

- TOS: 재단에서 발행하는 주 토큰입니다.
- LTOS: TOS를 stake하여 받을 수 있는 토큰입니다. LTOS는 rebase 토큰으로 수량이 변경될 수 있습니다. 채권 구매 시 자동으로 TOS는 LTOS로 변환되며 채권의 표현 단위입니다.
- sTOS: TOS를 lock-up 기간과 함께 stake할 때 LTOS와 함께 sTOS를 받습니다. (lock-up은 LockTosV2 컨트랙트가 구현하고 있으며 이번 코드 감사의 범위에 포함되지 않습니다).

StakingV2

- 사용자는 보유한 TOS를 stake하여 LTOS를 받을 수 있습니다.
- 사용자는 보유한 TOS를 lock-up을 기간을 포함하여 stake하여 LTOS와 sTOS를 받을 수 있습니다.
- LTOS는 rebase token으로 시간이 지날수록 수량이 늘어납니다.
- 사용자는 보유한 LTOS를 TOS로 청구할 수 있습니다.

BondDepository

- 관리자는 채권 시장을 생성할 수 있습니다. 현재 지원하는 시장은 ETH만을 지원합니다.
- 사용자는 채권 시장에서 채권을 구매할 수 있습니다. 채권은 LTOS 형태로 지급되며 LTOS 수량에 비례하여 Treasury에 TOS가 발행됩니다.

Treasury

- 사용자가 채권을 구매할 때 지불한 ETH와 발행된 채권으로 청구할 수 있는 TOS를 관리합니다.

Audit Summary

Delivery Date: 2022. 09. 29

Timeline: 2022. 09. 06 - 2022. 09. 26

Vulnerability Summary

Total Findings: 14

Severity	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Informational	7	0	0	0	0	0	7
Minor	3	0	0	0	0	0	3
Medium	3	0	0	0	0	0	3
Major	0	0	0	0	0	0	0
Critical	1	0	0	0	0	0	1

Files In Scope

- contracts/BondDepository.sol
- contracts/BondDepositoryProxy.sol
- contracts/BondDepositoryStorage.sol
- contracts/StakingV2.sol
- contracts/StakingV2Proxy.sol
- contracts/StakingV2Storage.sol
- contracts/TOSValueCalculator.sol
- contracts/Treasury.sol
- contracts/TreasuryProxy.sol
- contracts/TreasuryStorage.sol
- contracts/libraries/LibBondDepository.sol
- contracts/libraries/LibTreasury.sol
- contracts/libraries/LibStaking.sol
- contracts/common/ProxyAccessCommon.sol

- contracts/common/AccessRoleCommon.sol
- contracts/proxy/VaultProxy.sol
- contracts/proxy/VaultStorage.sol

List Of Findings

ID	Severity	Title	Status
TS2-1	Informational	Use require instead of modifier (non-exist market)	Resolved
TS2-2	Informational	Unnecessary Variable (id)	Resolved
TS2-3	Informational	Unnecessary if statement (isOpened)	Resolved
TS2-4	Informational	Variable Description Mismatch (_status)	Resolved
TS2-5	Informational	No Event Emitted (StakingV2)	Resolved
TS2-6	Critical	Potential Lose of LTOS	Resolved
TS2-7	Medium	Transfer LTOS instead of TOS (_claimAmount)	Resolved
TS2-8	Medium	Perpetual Bond	Resolved
TS2-9	Informational	Unnecessary if statement (getWETHPoolTOSPrice)	Resolved
TS2-10	Minor	OOG-friendly loop (backings)	Resolved
TS2-11	Informational	No Event Emitted (deleteBackingList)	Resolved
TS2-12	Minor	Function Implementation Mismatch (requestMint)	Resolved
TS2-13	Minor	Implicit Admin Role (MITNER_ROLE, BURNER_ROLE, POLICY_ROLE)	Resolved
TS2-14	Medium	Use DEFAULT_ADMIN_ROLE as proxy admin	Resolved

TS2-1 - Use require instead of modifier (non-exist market)

ID	Type	Severity	Location	Status
TS2-1	Coding Style	Informational	contracts/BondDepository.sol#L116 contracts/BondDepository.sol#L132 contracts/BondDepository.sol#L149	Resolved

Description

market의 존재 여부를 판단하는 require를 동일하게 여러번 사용하고 있습니다.

Recommendation

소스 코드의 가독성과 유지보수의 편의성을 위하여 하나의 modifier를 사용하기를 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 4ec020114302447d50183f1b50366c21ada44f31

TS2-2 - Unnecessary Variable (id)

ID	Type	Severity	Location	Status
TS2-2	Gas Optimization	Informational	contracts/BondDepository.sol#L216 contracts/BondDepository.sol#L242	Resolved

Description

_id 변수가 함수 파라미터로 이미 존재함에도 불구하고 동일한 값으로 id 변수를 선언하고 있습니다. 코드 가독성과 불필요한 메모리 낭비를 줄이기 위해 _id 변수를 제거하는 것을 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- a3451d084f2db4ed9a2b02cbc264bc6097578ae4
- a55f084d6617b8d1d39f818370005aa2c00bc0db

TS2-3 - Unnecessary if statement (isOpened)

ID	Type	Severity	Location	Status
TS2-3	Coding Style	Informational	contracts/BondDepository.sol#L366-L370	Resolved

Description

조건이 참일 경우 true를 반환하는 if문은 불필요합니다.

Recommendation

아래 코드로 if문을 생략할 수 있습니다.

```
return block.timestamp < markets[_marketId].endSaleTime && markets[_marketId].capacity > 0;
```

Alleviations

아래 커밋에서 반영되었습니다.

- 5326ffdcdcb3f93dfff5cae0e002181155854d0

TS2-4 - Variable Description Mismatch (_status)

ID	Type	Severity	Location	Status
TS2-4	Coding Style	Informational	contracts/BondDepositoryStorage.sol#L40-L50	Resolved

Description

_status 변수의 조작 과정을 설명하는 주석이 _notEntered로 잘못 기술되어있습니다. _notEntered는 OpenZeppelin의 이전 버전 ReentrancyGuard 구현에서 사용하던 boolean 자료형 변수입니다. 구현을 올바르게 설명하기 위하여 uint256 _status에 맞게 주석을 변경하는 것을 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- c96b90892bef301592a26c0f92320d37e3d4f097

TS2-5 - No Event Emitted (StakingV2)

ID	Type	Severity	Location	Status
TS2-5	Coding Style	Informational	contracts/StakingV2.sol#L66-L99	Resolved

Description

setAddressInfos, setRebasePerEpoch, setIndex, setBasicBondPeriod 함수들은 시스템의 중요한 상태 변수 tos, lockTOS, treasury, rebasePerEpoch, index_, basicBondPeriod를 변경합니다. 사용자에게 이러한 변경들을 알릴 수 있도록 이벤트를 발생시키는 것을 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 0f521cae03aae4a760a2258b9da36022bbc235ca

TS2-6 - Potential Lose of LTOS

ID	Type	Severity	Location	Status
TS2-6	Logical Issue	Critical	contracts/StakingV2.sol#L168-L184 contracts/StakingV2.sol#L792-L814	Resolved

Description

StakingV2 컨트랙트의 stake 함수는 유저가 TOS를 stake하고 LTOS를 받도록 구현되어잇습니다. 이 때 해당 staking을 구분하는 stakeId는 userStakings[msg.sender][1]를 사용하며 이는 _checkStakeId 함수에서 stakingIdCounter+1으로 할당됩니다. 이후 _createStakeInfo 함수를 호출해 allStakings[_stakeId] 상태 변수에 유저의 LTOS 수량을 기록합니다.

만약 동일한 유저가 stake 함수를 여러번 호출한다면 매 호출마다 TOS는 Treasury에 전달되지만 고정된 stakeId를 이용해 _createStakeInfo 함수를 호출하기에 됩니다. _createStakeInfo 함수는 allStakings[_stakeId]의 존재 유무를 고려하지 않고 일방적으로 할당하기 때문에 해당 유저의 allStakings[_stakeId].ltos 는 가장 마지막으로 stake 함수를 호출할 때의 값으로 결정됩니다.

따라서 여러번 stake함수를 호출한 유저는 전송한 TOS의 총 수량에 대한 LTOS가 아닌 마지막에 전송한 TOS 수량에 대하여 LTOS를 받게 됩니다.

Recommendation

아래와 같은 방식을 이용해 유저가 stake를 여러번 호출하는 것을 막거나 여러번 호출할 경우 LTOS를 증가시키도록 권장합니다.

1. _createStakeInfo 함수에서 allStakings[_stakeId]가 이미 존재한다면 트랜잭션을 revert합니다.
2. _createStakeInfo 함수에서 allStakings[_stakeId] 존재 여부에 따라 LibStaking.UserBalance의 deposit, ltos 값들을 증감합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 209b9e430ed22cb2c74f9164b719b3a485c63572

TS2-7 - Transfer LTOS instead of TOS (_claimAmount)

ID	Type	Severity	Location	Status
TS2-7	Volatile Code	Medium	contracts/StakingV2.sol#L282 contracts/StakingV2.sol#L393	Resolved

Description

resetStakeGetStosAfterLock 함수는 다음과 같은 형식으로 overloading 되어있습니다.

1. function resetStakeGetStosAfterLock(uint256 _stakeId, uint256 _addAmount, uint256 _claimAmount, uint256 _periodWeeks) external;
2. resetStakeGetStosAfterLock(uint256 _stakeId, uint256 _addAmount, uint256 _periodWeeks) external;
3. resetStakeGetStosAfterLock(uint256 _stakeId, uint256 _claimAmount) external;

이 중 _claimAmount 인자가 포함된 1, 3번 형식은 주석에 따르면 사용자가 청구할 LTOS 수량입니다.

resetStakeGetStosAfterLock 함수에서 Treasury 컨트랙트의 requestTransfer 함수를 호출하는데 이 때 인자로 _claimAmount를 사용합니다. requestTransfer 함수의 _amount 인자는 주석에 따르면 전송할 TOS 수량이기 때문에 LTOS 수량을 의미하는 _claimAmount를 사용할 경우 TOS 수량이 의도한 수량보다 더 적거나 더 많이 전송될 수 있습니다.

Recommendation

LTOS 수량을 TOS 수량으로 변환해주는 getLtosToTos 함수의 반환값을 requestTransfer의 인자로 사용하길 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- c13cad4067b1793cb2c6f677133358eb992179ed

TS2-8 - Perpetual Bond

ID	Type	Severity	Location	Status
TS2-8	Logical Issue	Medium	contracts/StakingV2.sol#L41-L948	Resolved

Description

StakingV2 컨트랙트가 LTOS 형태로 구현하는 채권 기능은 lock-up 기간이 존재하지만 실질적으로는 lock-up 종료 기간 이후에 청구할 경우 추가적인 기간만큼 이자를 청구받을 수 있는 영구채입니다.

Recommendation

구현하려는 채권 기능이 "lock-up 종료 기간이후 청구할 수 있는 영구채"가 아닐 경우 이자 지급 시점을 lock-up 종료 기간까지로 제한하는 것을 권장합니다.

Alleviations

개발팀은 구현하려는 채권 기능이 "lock-up 종료 기간 청구할 수 있는 영구채"가 맞다고 응답했습니다.

TS2-9 - Unnecessary if statement (getWETHPoolTOSPrice)

ID	Type	Severity	Location	Status
TS2-9	Coding Style	Informational	contracts/TOSValueCalculator.sol#L108-L117	Resolved

Description

getWETHPoolTOSPrice 함수의 첫 번째 if문의 조건은 절대 만족될 수 없으며 tosOrder가 0도 1도 아니면 항상 두 번째 if문의 else로 넘어가게 됩니다.

Recommendation

코드 가독성과 불필요한 연산을 제거하여 gas 소비를 줄이기 위하여 첫 번째 if문을 제거하길 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 1f94371326acbbf242702da943777c37d7b0b0b8

TS2-10 - OOG-friendly loop (backings)

ID	Type	Severity	Location	Status
TS2-10	Gas Optimization	Minor	contracts/Treasury.sol#L163-L197	Resolved

Description

_addBackingList, deleteBackingList 함수에서 array 타입인 상태 변수 backings를 순회하여 _address의 존재 여부를 판정하거나 인덱스를 찾습니다. backings 변수는 길이가 제한되어있지 않은 배열이기에 backings 배열의 길이가 길 경우 반복문에서 out-of-gas 에러가 발생할 수 있습니다.

Recommendation

mapping(address => uint256) backingIndexPlusOne 같은 상태 변수를 추가하여 backings 배열의 인덱스를 저장한다면 1개의 스토리지 슬롯을 추가로 사용하는 대신 두 함수에서 반복문을 제거할 수 있습니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 7fe0d32b5309b483989190aaf7cfbe1dee4ee8bb
- e2885fdd88e1a8b6e63e626c7f4e569cbd7de4b3

TS2-11 - No Event Emitted (deleteBackingList)

ID	Type	Severity	Location	Status
TS2-11	Coding Style	Informational	contracts/Treasury.sol#L181-L197	Resolved

Description

deleteBackingList 함수는 _addBackingList 함수와 대응하는 관계로서 Treasury 컨트랙트가 지원하는 자산을 제거할 수 있는 함수입니다. 이는 시스템의 중요한 상태를 변경할 수 있는 함수이에 이벤트를 발생시키는 것을 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- 2af4e050ffcef6d3eb84ab3936b746209eb85c3a

TS2-12 - Function Implementation Mismatch (requestMint)

ID	Type	Severity	Location	Status
TS2-12	Logical Issue	Minor	contracts/Treasury.sol#L248-L261 contracts/interfaces/ITreasury.sol#L57-L60	Resolved

Description

requestMint 함수의 주석과 실제 구현이 일치하지 않습니다. 주석에선 TOS를 발행하고 recipient에게 전달한다고 설명하고 있지만 실제 구현에선 TOS를 Treasury 컨트랙트에게 발행하며 recipient에게 TOS를 전달하지 않습니다.

Recommendation

주석 내용과 함수 구현을 일치시키길 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- f3a64bf0328ee79bd693fb9112e3dcb13832e236
- 19bb9828b345e8f85935540e957dd9774258e15f

TS2-13 - Implicit Admin Role (MITNER_ROLE, BURNER_ROLE, POLICY_ROLE)

ID	Type	Severity	Location	Status
TS2-13	Logical Issue	Minor	contracts/common/AccessRoleCommon.sol#L4-L11 contracts/proxy/VaultProxy.sol#L21	Resolved

Description

AccessRoleCommon에 정의된 Role들은 다음과 같습니다:

- ADMIN_ROLE
- MINTER_ROLE
- BURNER_ROLE
- POLICY_ROLE

각 Role들의 Admin Role은 기본적으로 DEFAULT_ADMIN_ROLE이며 AccessControl의 _setRoleAdmin 함수를 이용하여 별도의 Admin Role을 정의할 수 있습니다. _setRoleAdmin 함수는 아래 코드에서 호출합니다:

- VaultProxy.sol#L21: _setRoleAdmin(ADMIN_ROLE, ADMIN_ROLE);

따라서 각 Role과 그에 대응하는 Admin Role은 다음과 같은 테이블로 정의할 수 있습니다:

Role	Admin Role
ADMIN_ROLE	ADMIN_ROLE
MINTER_ROLE	DEFAULT_ADMIN_ROLE
BURNER_ROLE	DEFAULT_ADMIN_ROLE
POLICY_ROLE	DEFAULT_ADMIN_ROLE

이는 아래와 같은 문제점을 야기합니다:

· MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role이 DEFAULT_ADMIN_ROLE이기 때문에 ADMIN_ROLE을 가지고 있더라도 DEFAULT_ADMIN_ROLE이 없는 경우 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE롤에 대해 grantRole, revokeRole을 수행할 수 없습니다.

· DEFAULT_ADMIN_ROLE을 부여할 경우 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role을 부여하게 부여하게 됩니다.

Recommendation

위와 같은 Role - Admin Role 관계가 의도된 것이 아니라면 DEFAULT_ADMIN_ROLE와 ADMIN_ROLE중 하나만 사용하여 Role - Admin Role 관계를 명확하게 정의하는 것을 권장합니다. 만약 ADMIN_ROLE과 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role을 별도로 구분해야 한다면 별도의 Role(e.g., MANAGER_ROLE)을 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role로 사용합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- eae67d53e458959b1b749b57982793b121a73ac1
- 3d168e216db4b2a9d548510d43bd8a53f6d28697

TS2-14 - Use DEFAULT_ADMIN_ROLE as proxy admin

ID	Type	Severity	Location	Status
TS2-14	Logical Issue	Medium	contracts/common/ProxyAccessCommon.sol#L27 contracts/common/ProxyAccessCommon.sol#L33 contracts/common/ProxyAccessCommon.sol#L43-L44 contracts/common/ProxyAccessCommon.sol#L95	Resolved

Description

DEFAULT_ADMIN_ROLE은 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role입니다. ProxyAccessCommon에서 Proxy Admin에 대한 접근 제한
을 DEFAULT_ADMIN_ROLE로 이용하게 될 경우 Proxy Admin
은 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE을 제어할 수 있는 권한도 가지게 됩니다.

Recommendation

Proxy Admin이 MINTER_ROLE, BURNER_ROLE, POLICY_ROLE의 Admin Role로 정의하는 것이 의도된 것이 아니라면 별도의 PROXY_ADMIN_ROLE을 정의하는 것을 권장합니다.

Alleviations

아래 커밋에서 반영되었습니다.

- c13cad4067b1793cb2c6f677133358eb992179ed

Finding Categories

Gas Optimization

기존 소스 코드와 논리적으로 동일하게 동작하지만 EVM opcodes를 최적화하여 컨트랙트 배포 혹은 실행 시 트랜잭션 수수료를 줄일 수 있습니다.

Control Flow

컨트랙트가 기능을 제한하는 것이 올바른지 고려합니다. 이는 소유권과 같이 특정 사용자만 수행할 수 있는 기능 등을 포괄합니다.

Volatile Code

특정 코드가 예상치 못한 행동을 하거나 취약점이 존재하는 특정 edge case를 고려합니다.

Language Specific

Solidity에서만 발생하는 이슈들을 고려합니다.

Coding Style

생성된 byte-code에는 영향을 끼치지 않지만 유지보수가 용의하도록 코드베이스를 수정하는 것을 고려합니다.