Security/Privacy/Scalability issues in Kyber Network: Joint Liquidity Mining Campaign with NFTs and ERC20 tokens as rewards.

Joel Ng, U1921224F
School of Computer Science and Engineering
Nanyang Technological University
JNG203@e.ntu.edu.sg

Tan Wen Kai, U1922904C
School of Computer Science and Engineering
Nanyang Technological University
WTAN155@e.ntu.edu.sg

Lee Xuanhui, U1921226K
School of Computer Science and Engineering
Nanyang Technological University
LEEX0054@e.ntu.edu.sg

Abstract—This paper talks about the security aspect of decentralized applications (dApp) which are smart contracts deployed on Ethereum blockchain, specifically Kyber contracts. By analyzing the smart contract's [1] code in Solidity, some tests were done to see if there are possible security exploits for Kyber Contracts that will result in monetary loss of the users entrusting the money with the dApp. In this paper, we talk about the possible security exploits and measures to prevent them.

I. INTRODUCTION

Blockchain is a distributed, immutable ledger that records transactions and tracks assets in a network. The goal of a blockchain is to allow digital information to be recorded and distributed, but not edited. Since Bitcoin, its first widespread use, the applications of blockchain technology have exploded - from smart contracts to decentralized finance and most recently, non-fungible tokens (NFT) [2], the outlook for blockchain technology is bright and exciting.

However, there are still concerns with regards to the security, privacy, and scalability of the blockchain. There have been many security scandals in the blockchain world with far reaching consequences costing users and investors millions of dollars around the world. A recent and famous example would be the hack on PolyNetwork where the hacker exploited a flaw in their smart contracts and stole about USD$600million worth of tokens [3]. Scaling is also a serious issue. Long transaction times and high fees are negatively affecting many blockchains right now, especially the largest one - Ethereum. Privacy is another major problem with many blockchains. The public nature of many blockchains provide opportunities for identification by tracing transactions - making them only pseudonymous.

Our group's chosen project topic is - Kyber Network: Joint Liquidity Mining Campaign with NFTs and ERC20 tokens as rewards.

II. MOTIVATION AND LITERATURE SURVEY

A. Motivation

The main focus for this project would be security as there have been many instances where security vulnerabilities [4] in dApps have been exploited resulting in up to hundreds of million dollars lost for the users.

B. Literature Survey

PancakeSwap, a popular dApp built on Binance Smart Chain had a vulnerability in their MasterChef contract that could lead to infinite minting of one of their tokens ($SYRUP).

The MasterChef contract is a contract where a user can deposit different tokens allowed by the developers in PancakeSwap in order to earn PancakeSwap's native token $CAKE.

A user can deposit $CAKE into their MasterChef contract by calling the enterStaking() method and get $SYRUP in return as a proof of depositing $CAKE into the contract. Once the user is done, the user can retrieve his deposited $CAKE and extra $CAKE rewards by calling the leaveStaking() method and this will burn $SYRUP.

However, there are many other withdrawal methods in the MasterChef contract that allows the user to withdraw their tokens. The emergencyWithdraw() method allows the user to withdraw their deposited tokens without getting any $CAKE rewards. However, it missed out a line of code and it does not burn the $SYRUP unlike the leaveStaking() method[5].

This is a security vulnerability and could be catastrophic if $SYRUP has economic value due to the fact that the user can keep calling enterStaking() to deposit $CAKE and call emergencyWithdraw() repeatedly to get an infinite supply of $SYRUP.

This example provides relevant information and lessons learnt for the chosen project topic as likewise, the project involves the user staking tokens by depositing tokens into the FairLaunch contract and receiving NFT tokens as rewards - similar to PancakeSwap.

III. OBSERVATIONS AND ANALYSIS

In this project, the following issues have been observed to be potentially catastrophic or of concern for the dApp's security.

A. Unauthorized users using KyberFairLaunch's AddPool() method

The KyberFairLaunch contract has an AddPool() method which is to create a reward pool with a token so that users can deposit the token and earn Kyber rewards the longer they deposit the token.

Only people with SpecialAccess should be allowed to call the function AddPool(). If anyone with bad intentions can get access to the AddPool() method, they are able to add any random ERC20 token and give the pool high rewards per block, resulting in hyperinflation and will destroy the Kyber tokenomics.

B. Issue of Overflow/Underflow of Token

Overflow occurs when a number gets incremented above its maximum value for an unsigned integer. Underflow occurs when a number gets decremented below its minimum value for an unsigned integer.

The NFT Tokens gained from the KyberFairLaunch contract are used to exchange for NFTs. It is possible that the number of NFT Tokens will overflow/underflow in the smart contract. In this scenario, an underflow bug is more detrimental because a user exploiting the underflow bug can max out her/her balance. Eg; If an exploiter only owns 999 tokens and tries to mint an NFT costing 1000 tokens, he will end up with a max balance of 2^256-1 tokens.

C. User Requires NFT Tokens to Mint NFT

To mint the Kyber NFT, users must have sufficient Kyber NFT Reward tokens to exchange for Kyber NFT. The Kyber NFT Reward token can only be gained from the KyberFairLaunch contract (A proof that the user deposits liquidity to Kyber).

IV. PROPOSED SOLUTIONS

In this section, propose potential solutions to address the issues that you found in your analysis earlier. These solutions may be inspired from the lectures, invited talks, related works, or any other instance of similar development projects.

A. Solution to Unauthorized users Using KyberFairLaunch's AddPool() method

Import and Implement OpenZeppelin's AccessControl Ownable contract [6]. Add onlyOwner modifier to functions that require special access. With onlyOwner modifier, only the owner can access the function.

B. Solution to Overflow/Underflow of Token

Use SafeMath.sol[7]. It is a well-known library used in many smart contracts where it can provide the basic arithmetic operations while checking preconditions and postconditions to check whether an overflow/underflow has occurred. If an overflow or underflow has been detected, SafeMath fails the transaction's execution and changes the status of the transaction to 'Reverted'.

C. Solution to user Requires NFT Tokens to Mint NFT

In the method where the NFT is minted, there can be a 'require' statement to check if the user has enough NFT Tokens. (require IERC20(Token).balanceOf(msg.sender) >= amount)

If the user does not have enough tokens, the 'require' statement will throw an error and the transaction will be reverted.

V. CONCLUSION

In conclusion, the security aspects of the smart contract are too important to neglect when there are millions of dollars involved. Contracts holding substantial amounts of monetary value should be audited before deploying on the blockchain due to the immutability of these contracts.

For a project which handles tens of millions of dollars, if there was a bug in the code, and it led to an exploiter to drain out all of the money. Users who entrusted their money to the dApp will lose faith in decentralized finance.
Hence, security is the most important aspect due to the exorbitant amount of money involved / economic value.

REFERENCES

[1] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper, vol. 151, no.

[2]. C. Jazmin Goodwin, "What is an NFT? Non-fungible tokens explained", *CNN*, 2021. [Online]. Available: https://edition.cnn.com/2021/03/17/business/what-is-nft-meaning-fe-series/index.html. [Accessed: 26- Nov- 2021]

[3] GitHub. 2018. Ethereum smart contract best practices, Known attacks, https://consensys.github.io/smart-contract-bestpractices/knownattacks/

[4] "Poly Network Hack", 2021. [Online]. Available: https://www.cnbc.com/2021/08/23/poly-network-hacker-returns-remaining-cryptocurrency.html. [Accessed: 26- Nov- 2021]

[4]"PancakeSwap MasterChef", 2021. [Online]. Available: https://bscscan.com/address/0x73feaa1ee314f8c655e354234017be2193c9e24e#code. [Accessed: 26- Nov- 2021]

[6] GitHub. 2021. Openzepellin-contracts-Ownable.sol. onlyOwner() https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/Ownable.sol

[7] GitHub. 2021. Openzepellin-contracts, SafeMath.sol https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/utils/math/SafeMath.sol