FUNDAMENTAL AND TECHNICAL AUDIT

TCG World

April 8, 2022

Table of Contents

Project Details

Project Name
TCG World

Project Type
BEP20 & ERC20 Token

Contract Address
BSC: 0xF73d8276C15Ce56b2f4AeE5920e62F767A7f3aEA

ETH: 0x0d31DF7dedd78649A14aAe62D99CcB23aBCC3A5A

Blockchain
BSC & Ethereum

Token Name
TCG World

Token Ticker
TCG2

Decimals
9

Project Website
tcg.world

This Audit Was Created On
April 8, 2022

Mission Overview

Real-World Mission
Yes

Realizable Mission
Yes

Innovative Idea
No

MISSION OVERVIEW

TCG World is a metaverse gaming platform built on Unity. Within the platform users can earn TCGcoin 2.0, hunt for NFT collectibles, and invest in virtual real estate. All of the items a user can own are delivered in the form of NFTs.

CURRENT PROGRESS

TCG's game was just recently launched into the alpha stage, which is an important step for projects in this sector. They also recently released their NFT project, TCG Dragon Cave Club. While the utility for this NFT collection is not yet available, in the future holders will have access to events within TCG World. The team has also been able to launch their coin, TCGcoin 2.0, on multiple chains with a unified liquidity pool.

CONCLUSION

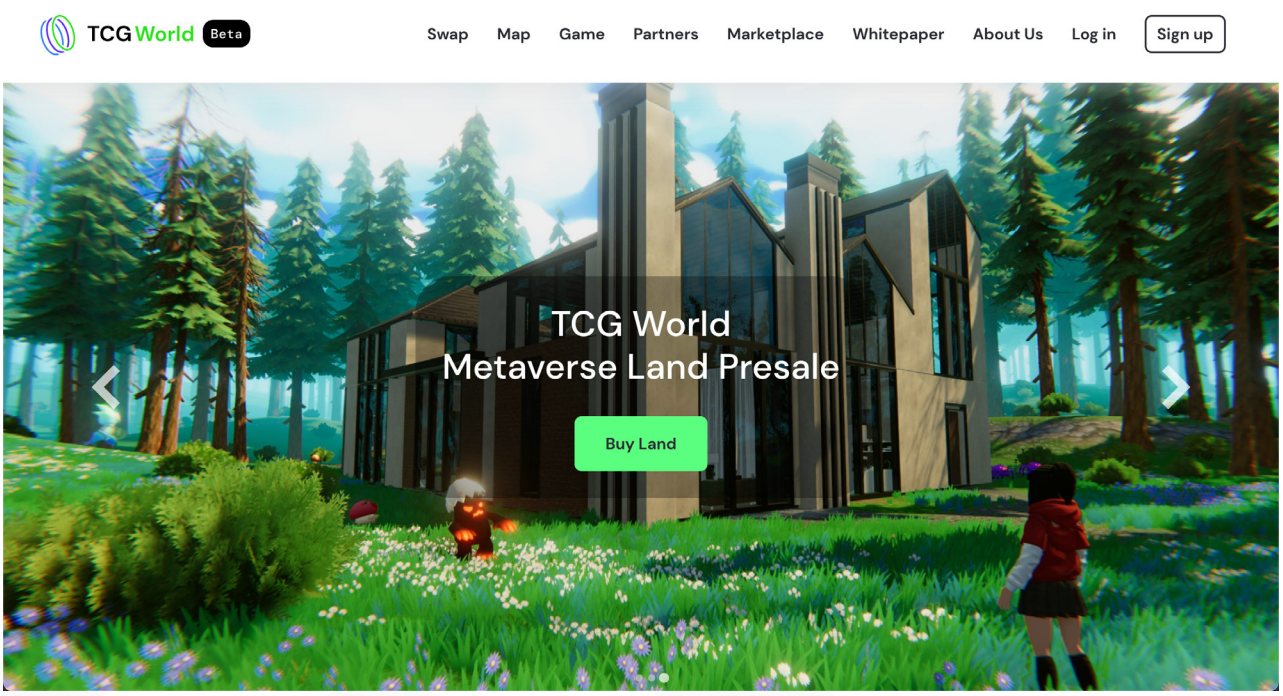
Through discussions with the team, it is apparent they are working diligently to create a game they can be proud of. In this saturated sector, many projects appear to rush the development of their games to get people onboarded as quickly as possible, but TCG has opted to take their time in crafting their game. Their project has instead focused on iterative building and attention to detail.

The game itself is being built on Unity, which is a platform mainly used for 3D mobile game development. Because the sector is so saturated, likely due to a recent spike in interest, it can be surmised that the details are what will help projects succeed while others become obsolete. While Unity is a good option for cross-platform gaming due to its flexibility, it does have limitations with textures and some other 3D elements where competitors may be more successful. Because of the shift to VR and Metaverse within the gaming community, it is not yet understood what role graphics and performance will play in the future. This makes it difficult to ascertain if the decision to build with Unity will hinder this project, or be something that the community doesn't require.

As a space sees more competition, it can be difficult to determine which companies and projects will succeed, so investments in sectors such as the metaverse tend to carry larger amounts of risk because investor sentiment is not clear. Because the metaverse is one of these sectors, it becomes difficult for all projects to reduce this risk, as we haven't seen the entry of large gaming studios which undoubtedly have more experience and resources. These unique situations increase the risk for investors in all projects within this sector, and are not specific to this particular project.

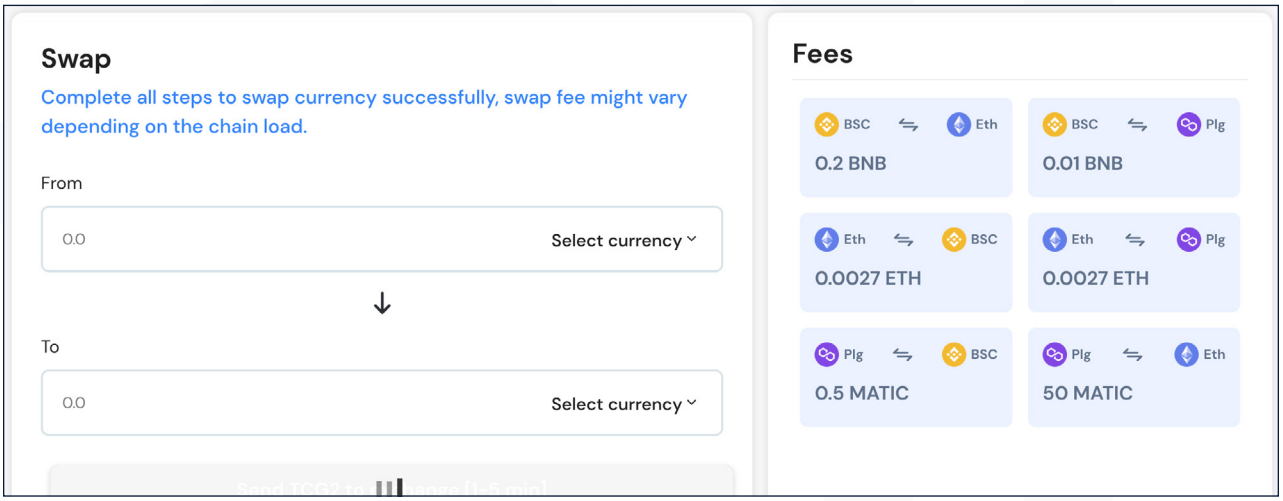
Design Audit

WEBSITE



Design Standards

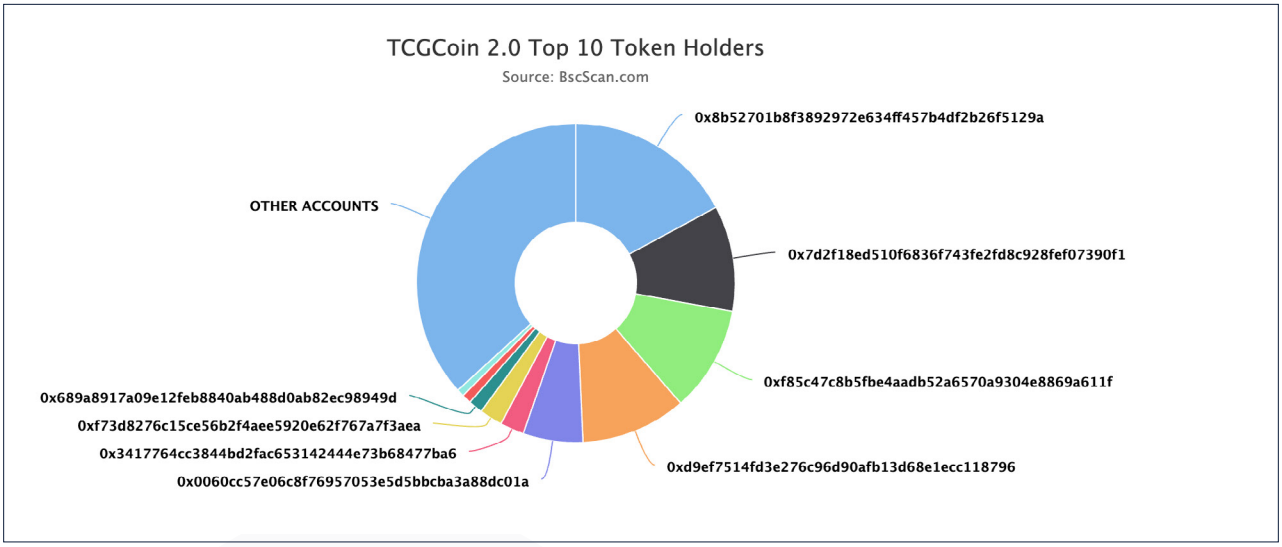
The cryptocurrency space is filled with dark theme communications materials, which presents a readability problem for those with visual impairments. TCG's decision to utilize a light theme for their website, swap, and other items improves legibility. The type treatments establish an understandable hierarchy which helps guide users through the experience to find the information they need efficiently.



SECURITY RISK ASSESSMENT: NONE

Top 10 Holders Overview

The top 10 holders collectively own 63.22% (177,026,960.42 Tokens).



1st address—Team Wallet
17.1429% of total supply.

2nd address—Team Wallet
10.7143% of total supply.

3rd address—Team Wallet
10.7143% of the supply.

4th address—Team Wallet
10.7143% of the supply.

5th address—Team Wallet
6.0787% of the supply.

6th address—Team Wallet
2.4357% of the supply.

7th address—TCGCoin 2.0 Smart Contract
2.271% of the supply.

Tokens from this address will be swapped for liquidity pool tokens once the contract owner enables the swapAndLiquify function.

8th address—Team Wallet
1.4619% of the supply.

9th address—PancakeSwap V2 Liquidity Pool
0.9461% of the supply.

10th address—MEXC Exchange Liquidity Pool
0.7439% of the supply.

CONCLUSION

At first, investors may be concerned that the team controls 8 out of 10 top wallets, but TCG World's team provided us with a full explanation of each wallet's allocation, including specifics on how tokens will be used in the game and future activities (in-game yield farming, staking, crystals hunting that will reward you with TCGCoin 2.0, marketing...). There is also a dedicated page on the TCG World whitepaper, which is accessible to the public via the website, that outlines token allocation. Our only advice and security concern is to transfer these tokens to a safer place, such as a multi-signature wallet. This would significantly reduce the possibility of a wallet owner getting hacked, which might cause significant damage to the token's price and project reputation.

Liquidity Pool Overview

Liquidity Locked
Yes

Locking Period
8 years (8/27/2030)

Auto-Liquidity Function
Yes

Total Value Locked (TVL)
$1,358,472 (74.1645% of liquidity pool)

Total Liquidity Value
4,351.77 WBNB ($1,836,769)

Liquidity Lock Platform
DeepLock.io

Liquidity Pool Holders
12

Liquidity Lock Address
0x3f4d6bf08cb7a003488ef082102c2e6418a4551e

Liquidity lock from DeepLock

CONCLUSION

Our team is happy that the TCG World team is committed to liquidity pool building, and that their whole token tax is allocated solely to the liquidity pool. This greatly aids the token's strength, and their liquidity pool is currently in fantastic condition. The fact that 74.16% of the liquidity pool is locked for the next eight years shows that the team is serious about the project's security and long-term viability. Our only recommendation is to lock the remaining 25.78% of liquidity to ensure that the project owner has no access to it.

2	0x0d4897e58374b8c8c12b949c84249184f84dad06	0.852564470230914914	25.7897%

Tokenomics Overview

Total Supply
280,000,000

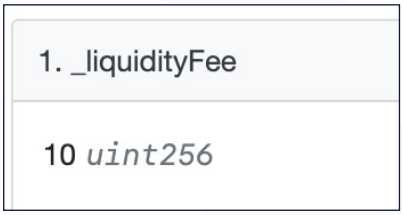
Tax Applied on Every Transaction
Not applied on wallet-to-wallet transfer

Tax Distribution

- 10% liquidity pool tax on sell

CONCLUSION

TCG Coin 2.0 currently has a 10% tax on sales during the development phase, however this tax is expected to be reduced over time and eventually become a tax-free token. In this section of the audit, we found no flaws or security concerns, and we like the way technical parts of tokenomics are evolving. We were one of the teams who audited the TCGCoin 2.0 smart contract before it was migrated and deployed on the main net, and we feel that a low token supply is more attractive and professional.



Liquidity Pool Tax on Sell

Team Overview

Core Team Members
18

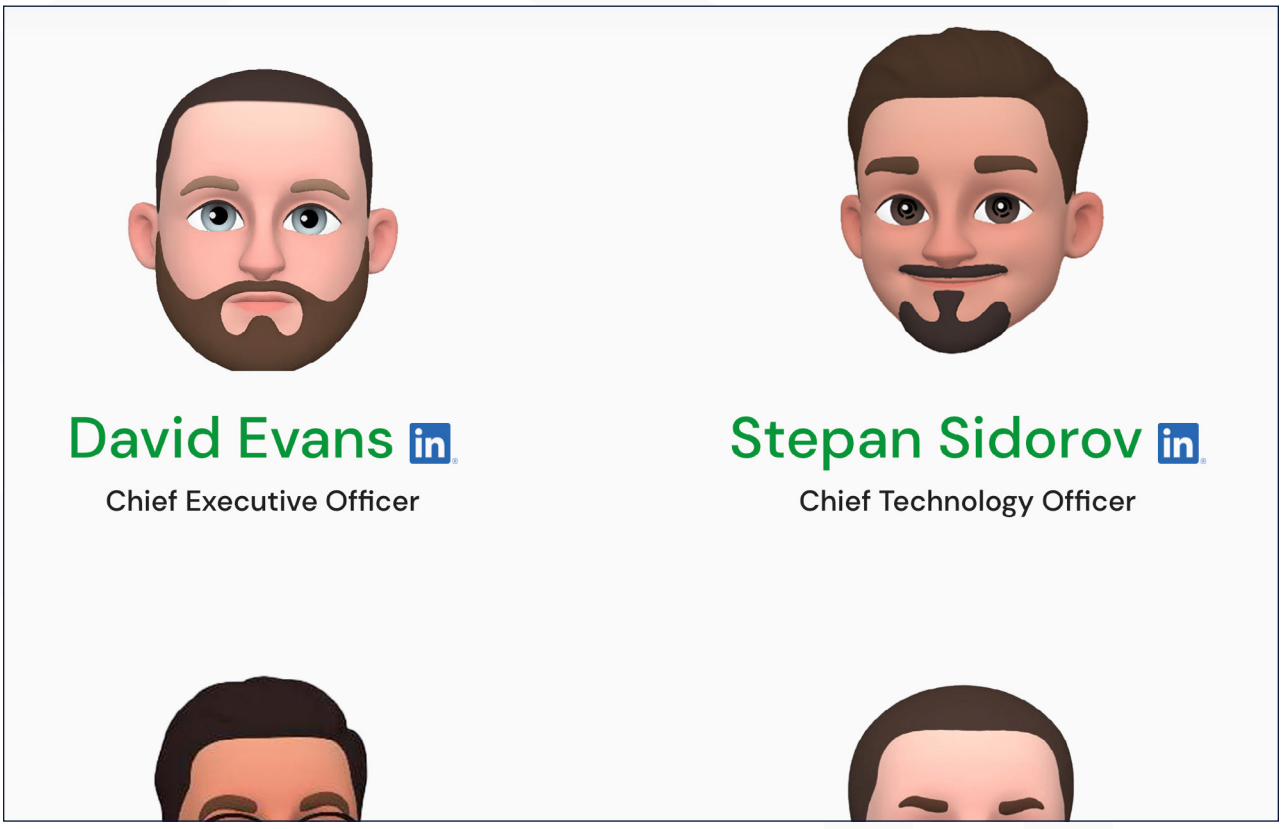
Doxxed Team
Yes

Existing AMAs
Yes

Team Members Have Previous Experience:
Yes

CONCLUSION

The TCG World team is entirely made up of professionals; each team member has prior expertise in the sector in which they work, and each team member's LinkedIn profile can be seen on the website. We met with the CEO of TCG World several times, and they always supplied us with the information we required, whether it was a response to a query or an explanation for our needs.



Technical Audit Overview

GENERAL ISSUES

- Security Issues: **PASSED**
- Gas & Fees Issues: **PASSED** *(FIXED 1 LOW SEVERITY ISSUE)*
- ERC Errors: **PASSED** *(FIXED 1 LOW SEVERITY ISSUE)*
- Compilation Errors: **PASSED**
- Design Logic: **PASSED**
- Timestamp Dependence: **PASSED**

SECURITY AGAINST CYBER ATTACKS

- Private user's data: **SECURED**
- Reentrancy: **SECURED**
- Cross-Function Reentrancy: **PASSED**
- Front Running: **PASSED**
- Taxonomy Attacks: **PASSED**
- Integer Overflow and Underflow: **PASSED**
- DoS (Denial of Service) with Unexpected Revert: **PASSED**
- DoS (Denial of Service) with Block Gas Limit: **PASSED**
- Insufficient Gas Griefing: **PASSED**
- Forcibly Sending ETH to a Contract: **PASSED**

In-Depth Analysis

GAS & FEES ISSUES

UPDATE [PASSED]: ***The developers included the "_maxLoopCount" variable in order to prevent the uncontrolled iterations.***

(BEFORE UPDATE)

We observed that the following functions:

- *function_getCurrentSupply() private view returns(uint256, uint256)*
- *function includeInReward(address account) external onlyOwner()*

Because FOR loops with dynamic arrays are used, the number of iterations is uncontrolled.

ERC ERRORS

UPDATE [PASSED]: ***"public" state was added***

(BEFORE UPDATE)

State variable visibility is not set for:

- *bool isIntoLiquidifySwap*

SECURITY ISSUES

SECURITY ISSUES: 0

We observed that the following functions:

- function lock(uint256 time) public virtual onlyOwner
- function unlock() public virtual
- function swapTokensForBNB(uint256 tokenAmounts) private
- function addLiquidity(uint256 tokenAmount, uint256 bnbAmount) private

are utilizing block.timestamp variable. That means the miner has the ability to "choose" the block.

To a certain level, a timestamp can be used to alter the outcome of a transaction in a mined block.

However, because those functions are private, can be called only by the contract owner, and do not generate sensitive information, the functions listed above are safe and secure.

Technical Audit Details

FUNCTIONS THAT CAN BE CALLED BY OWNER

Exclude/Include Address from Fees and/or Rewards::

- excludeFromFee
- excludeFromReward
- includeInFee
- includeInReward

Ownership

- lock -Renounce to ownership for a period of time ;
- renounceOwnership;
- unlock - Will be automatically called when the unlock time will arrive ‘
- transferOwnership - transfer the ownership to another address;

Set Fees

- setLiquidityFeePercent
- setTaxFeePercent
- setTaxFeeFlag - indicates if fee should be deducted from transfer;

Other Functions

- rescueBNBFromContract - rescue BNB sent by mistake directly to the contract;
- setSwapAndLiquifyEnabled - enable or disable auto LP;
- setMaxTxPercent - set the maximum value of transactions;

Penetration Testing

RE-ENTRANCY

What is Re-Entrancy?

A re-entrancy attack can arise when you write a function that calls another untrusted contract before resolving any consequences. If the attacker has authority over the untrusted contract, he can initial a recursive call back to the original function, repeating interactions that would otherwise not have occurred after the effects were resolved.

Attackers can take over the smart contract's control flow and make modifications to the data that the calling function was not anticipating.

To avoid this, make sure that you do not call an external function until the contract has completed all of the internal work.

TEST: PASSED

CROSS-FUNCTION RE-ENTRANCY

What is Cross-Function Re-Entrancy?

When a vulnerable function shares the state with another function that has a beneficial effect on the attacker, this cross-function re-entrancy attack is achievable. This re-entrancy issue that is the employment of intermediate functions to trigger the fallback function and a re-entrancy attack is not unusual.

Attackers can gain control of a smart contract by calling public functions that use the same state/variables as "private" or "onlyOwner" functions.

To avoid this, make sure there are no public functions that use private variables, and avoid calling routines that call external functions or use mutex (mutual exclusion).

TEST: PASSED

PENETRATION TESTING (CONTINUED)

FRONT RUNNING

What is Front Running?

Front running indicates that someone can obtain prior information of transactions from other beneficial owners by technology or market advantage, allowing them to influence the price ahead of time and result in economic benefit, which usually results in loss or expense to others

Since all transactions are visible in the block explorer for a short period of time before they are executed, network observers can see and react to an action before it is included in a block.

Attackers can front run transactions because every transaction is visible to the blockchain, even if it is in the “processing” or “indexing” state. This is a very low security vulnerability because it is based on the blockchain rather than the contract.

The only possible attack is seeing transactions made by bots. Using transaction fees, you can avoid bots.

TEST: PASSED

TAXONOMY ATTACKS

These taxation attacks can be made in 3 ways:

1. Displacement

Performed by increasing the gasPrice higher than network average, often by a multiplier of 10.

2. Insertion

Outbidding transaction in the gas price auction

3. Suppression (Block Stuffing)

The attacker sent multiple transactions with a high gasPrice and gasLimit to custom smart contracts that assert to consume all the gas and fill up the block’s gasLimit.

This type of attack occurs mainly for exchanges, so this smart contract is secured

TEST: PASSED

PENETRATION TESTING (CONTINUED)

INTEGER OVERFLOW AND UNDERFLOW

Overflow

An overflow occurs when a number gets incremented above its maximum value.

In the audited contract: uint8 private _decimals = 9;

(_decimals can't reach a value bigger than it's limit)

TEST: PASSED

Underflow

An overflow occurs when a number gets decremented below its maximum value.

(There are no decrementation functions for parameters and users can't call functions that are using uint values);

TEST: PASSED

Conclusion

This contract uses the updated version of SafeMath for Solidity 0.8 and above that will prevent Integer Overflows and Underflows.

DOS (DENIAL OF SERVICE) WITH UNEXPECTED REVERT

DoS (Denial of Service) attacks can occur in functions when you attempt to transmit funds to a user and the functionality is dependent on the successful transfer of funds.

This can be troublesome if the funds are given to a bad actor's smart contract (when they call functions like "Redeem" or "Claim"), since they can simply write a fallback function that reverts all payments.

There are no functions that deliver money to users, so attackers are unable to communicate using a contract with fallBack functions.

TEST: PASSED

PENETRATION TESTING (CONTINUED)

DOS (DENIAL OF SERVICE) WITH BLOCK GAS LIMIT

Each block has an upper bound on the amount of gas that can be spent, and thus the amount of computation that can be done. This is the Block Gas Limit. If the gas spent exceeds this limit, the transaction will fail. This leads to a couple possible Denial of Service vectors.

TEST: PASSED

INSUFFICIENT GAS GRIEFING

This attack can be carried out against contracts that accept data and use it in a sub-call on another contract.

This approach is frequently employed in multisignature wallets and transaction relayers. If the sub-call fails, either the entire transaction is rolled back or execution is resumed.

TEST: PASSED

Users can't execute sub-calls.

FORCIBLY SENDING ETH TO THE SMART CONTRACT

TEST: PASSED

This contract has a function that returns the BNB provided to it.

This audit was created by

Accepting a project audit can be viewed as a sign of confidence and is typically the first indicator of trust for a project, but it does not guarantee that a team will not remove all liquidity, sell tokens, or engage in any other type of fraud. There is also no method to restrict private sale holders from selling their tokens. It is ultimately your obligation to read through all documentation, social media posts, and contract code for each particular project in order to draw your own conclusions and define your own risk tolerance.

DRIVENsecurity accepts no responsibility for any losses or encourages speculative investments. This audit's material is given solely for information reasons and should not be construed as investment advice.