CERTIK

Code Security Assessment

Head Football

Jan 14th, 2022

Table of Contents

Summary

This report has been prepared for Head Football to discover issues and vulnerabilities in the source code of the Head Football project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Head Football
Platform	other
Language	Solidity
Codebase	https://github.com/HeadFootball/HeadFootball/tree/8d18fce9ba55c0be3d908267c1cfc00c64e07e79
Commit	8d18fce9ba55c0be3d908267c1cfc00c64e07e79

Audit Summary

Delivery Date	Jan 14, 2022
Audit Methodology	Static Analysis, Manual Review

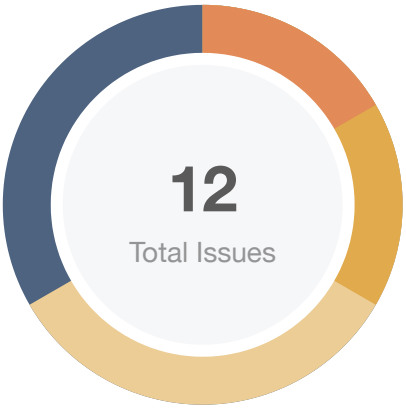
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	2	0	0	2	0	0
Medium	2	0	0	1	0	1
Minor	4	0	0	2	0	2
Informational	4	0	0	1	0	3
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
HEA	HEADFOOTBALL.SOL	35a168418a67bd23147bb05c1999687bec1aed26ba6b1775833c95279f3c724e

Findings



Critical	**0** (0.00%)
Major	**2** (16.67%)
Medium	**2** (16.67%)
Minor	**4** (33.33%)
Informational	**4** (33.33%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
GLOBAL-01	Third Party Dependencies	Volatile Code	Minor	Acknowledged
GLOBAL-02	Variable Could be Declared as `constant`	Gas Optimization	Informational	Resolved
GLOBAL-03	Missing Event Emitting	Coding Style	Informational	Acknowledged
GLOBAL-04	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
GLOBAL-05	Tokenomics	Control Flow	Medium	Acknowledged
HEA-01	Token Minted To Centralized Address	**Centralization / Privilege**	**Major**	Acknowledged
HEA-02	Incorrect Error Message	Logical Issue	Minor	Resolved
HEA-03	Updating `_uniswapV2Pair` Without Checking Existence	Volatile Code	Medium	Resolved
HEA-04	Lack of Zero Address Validation	Volatile Code	Minor	Resolved
HEA-05	Redundant Code	Logical Issue	Informational	Resolved
HEA-06	Potential Sandwich Attacks	Logical Issue	Minor	Acknowledged
HEA-07	Updating `_tOwned` Directly	Logical Issue	Informational	Resolved

GLOBAL-01 | Third Party Dependencies

Category	Severity	Location	Status
Volatile Code	Minor	Global	Acknowledged

Description

The contract is serving as the underlying entity to interact with third-party protocols. The scope of the audit treats 3rd party entities as black boxes and assumes their functional correctness. However, in the real world, 3rd parties can be compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties can possibly create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Recommendation

We understand that the business logic of this project requires interaction with the third-party `Swap` protocol. We encourage the team to constantly monitor the statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Alleviation

The client acknowledged.

GLOBAL-02 | Variable Could be Declared as `constant`

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

Variables `_tTotal`, `_name`, `_symbol` and `_decimals` could be declared as `constant` since these state variables are never to be changed.

Recommendation

We recommend declaring those variables as `constant`.

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

GLOBAL-03 | Missing Event Emitting

Category	Severity	Location	Status
Coding Style	Informational	Global	Acknowledged

Description

In the contract `HeadFootBall`, there are a bunch of functions can change state variables. However, these function do not emit event to pass the changes out of chain.

Recommendation

Recommend emitting events, for all the essential state variables that are possible to be changed during runtime.

Alleviation

The client acknowledged.

GLOBAL-04 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Acknowledged

Description

In the contract `HeadFootBall`, the role `owner` has the authority over the following function:

- excludeFromReward()
- includeInReward()
- excludeFromFee()
- includeInFee()
- setTaxFeePercent()
- setMarketingFeePercent()
- setMinNumTokensSellToGetBnb()
- updateRouter()
- setMaxTxAmount()
- setSwapAndLiquifyEnabled()
- setMarketingWallet()
- claimStuckTokens()

In the contract `Ownable`, the role `owner` has the authority over the following function:

- renounceOwnership()
- transferOwnership()

Additionally, `_marketingWallet` will be used to receive `BNB`.

Any compromise to these accounts may allow the hacker to manipulate the project through these functions.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles. OR
- Remove the risky functionality.

Alleviation

The client acknowledged.

GLOBAL-05 | Tokenomics

Category	Severity	Location	Status
Control Flow	Medium	Global	Acknowledged

Description

The HeadFootBall Protocol is a decentralized finance (DeFi) token. Each HeadFootBall transaction is taxed taxFee/MarketingFee fees totalling 6% of the transaction amount. The first fee is redistributed to all existing holders using a form of rebasing mechanism whilst the other 4% is accumulated internally until a sufficient amount of capital has been amassed to perform a swap. When this number is reached, the total tokens accumulated will be converted to BNB and transferred to a `_marketingWallet`. The owner can updated the fee rates at any time.

Recommendation

We recommend to publish this feature to the community.

Alleviation

The client acknowledged.

HEA-01 | Token Minted To Centralized Address

Category	Severity	Location	Status
Centralization / Privilege	**Major**	HEADFOOTBALL.SOL: 858~859	Acknowledged

Description

The total supply amount of tokens that are minted to the centralized address `_msgSender()` who is `owner`, may raise the community's concerns about the centralization issue.

Recommendation

We advise the client to carefully manage the `owner` account's private key and avoid any potential risks of being hacked. We also advise the client to adopt Multisig, Timelock, and/or DAO in the project to manage this specific account in this case.

Alleviation

The client acknowledged.

HEA-02 | Incorrect Error Message

Category	Severity	Location	Status
Logical Issue	Minor	HEADFOOTBALL.SOL: 977	Resolved

Description

The error message in `require(_isExcluded[account], "Account is already excluded")` does not describe the error correctly.

Recommendation

The message "Account is already excluded" can be changed to "Account is not excluded" .

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

HEA-03 | Updating `_uniswapV2Pair` Without Checking Existence

Category	Severity	Location	Status
Volatile Code	Medium	HEADFOOTBALL.SOL: 1013~1020	Resolved

Description

In function `updateRouter()`, new pair is created without checking existence. If the new pair is already created, `IUniswapV2Factory` will revert the transaction. As a result, the new router will never be set successfully.

Recommendation

We recommend the client to change as below:

```
function updateRouter(address newAddress) external onlyOwner {
    require(newAddress != address(uniswapV2Router), "TOKEN: The router already has that
address");
    uniswapV2Router = IUniswapV2Router02(newAddress);
    address get_pair =
IUniswapV2Factory(uniswapV2Router.factory()).getPair(address(this),
uniswapV2Router.WETH());
    if (get_pair == address(0)) {
        uniswapV2Pair =
IUniswapV2Factory(uniswapV2Router.factory()).createPair(address(this),
uniswapV2Router.WETH());
    } else {
        uniswapV2Pair = get_pair;
    }
}
```

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

HEA-04 | Lack of Zero Address Validation

Category	Severity	Location	Status
Volatile Code	Minor	HEADFOOTBALL.SOL: 1030	Resolved

Description

The `_marketingWallet` lacks of zero address validation.

Recommendation

We advise the client to add an input validation in `setMarketingWallet()` as follows.

```
require(marketingWallet != address(0), "marketingWallet address can not be zero!");
```

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

HEA-05 | Redundant Code

Category	Severity	Location	Status
Logical Issue	Informational	HEADFOOTBALL.SOL: 1224~1225	Resolved

Description

The condition `!_isExcluded[sender] && !_isExcluded[recipient]` can be included in `else` .

Recommendation

The following code can be removed:

```
... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
        _transferStandard(sender, recipient, amount);
    } ...
```

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

HEA-06 | Potential Sandwich Attacks

Category	Severity	Location	Status
Logical Issue	Minor	HEADFOOTBALL.SOL: 1203~1204	Acknowledged

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by backrunning (after the transaction being attacked) a transaction to sell the asset.

The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- `uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens()`

Recommendation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

Alleviation

The client acknowledged.

HEA-07 | Updating `_tOwned` Directly

Category	Severity	Location	Status
Logical Issue	Informational	HEADFOOTBALL.SOL: 1098	Resolved

Description

The function `_takeMarketing()` updates `_tOwned[address(this)]` without checking whether `address(this)` is excluded from reward or not.

Recommendation

We recommend the client to update `_tOwned[address(this)]` only when `address(this)` is excluded from reward.

Alleviation

The team heeded our advice and changed related codes. Code change was applied in commit 2bc183d1fb7e411942880aa53f54b90c58ab4537.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is 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 assessment. This report does 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.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.