SOLIDITY FINANCE

Wattbit

Smart Contract Audit Report

AUDIT SUMMARY

Wattbit, Inc. is building a platform in which users can purchase TWL NFTs, receive Badges for holding TWL NFTs, and use Badges to earn commission on Sponsor NFT sales.

For this audit, we reviewed the following contracts on the Rinkeby Testnet:

- Badge contract at 0x36a21e0F3ef268de1b17eC7F37a7482c2cE32388.
- CBB contract at 0xfbc78d985185A6FF1ae98b7D53ffAf7C4cFD25A6.
- Sponsor contract at 0xec1c44384bC6D40E99B6a438A49163da3bc86BF4.

AUDIT FINDINGS

Please ensure trust in the team prior to investing as they have some control in the ecosystem.
Date: February 8th, 2022.
Updated: February 21st, 2022 to include resolutions to various findings.

Finding #1 - CBB - Medium (Resolved)

Description: *The mintPresale() function is vulnerable to reentrancy attacks although it is limited to only whitelisted users. Reentrancy is made possible via the call to _safeMint(), which requires that the receiver has implemented the onERC721Received() function in the event that the receiver is a contract.*
Risk/Impact: *Whitelisted users can use the onERC721Received() function to execute arbitrary logic and reenter the mintPresale() function in an effort to mint NFTs at the presale price in excess of the individual presale limit and even the maximum supply limit.*
Recommendation: *The logic in the mintPresale() function should be restructured to follow the Checks-Effects-Interactions pattern. The* `presaleMints[msg.sender]` *value should be increased by the appropriate amount prior to minting any NFTs.*
Resolution: *The team has implemented the above recommendation.*

Finding #2 - CBB & Sponsor - Low (Resolved)

Description: *The reserve() function is vulnerable to reentrancy attacks although it is limited to situations in which the owner is a contract address.*
Risk/Impact: *The owner can use this function to perform a reentrancy attack in an effort to mint NFTs in excess of*

the maximum reserved amount.
Recommendation: *The logic in the reserve() function should be restructured to follow the Checks-Effects-Interactions pattern. The* `reserved` *value should be increased by the appropriate amount prior to minting any NFTs.*
Resolution: *The team has implemented the above recommendation.*

Finding #3 - CBB & Sponsor - Low

Description: *Any excess ETH supplied to the contract during minting is not returned to the user.*
Risk/Impact: *Users will lose any excess funds sent as payment.*
Recommendation: *The contract should require the user supplies the exact amount of ETH needed to mint the desired amount of NFTs.*

Finding #4 - Badge - Low (Resolved)

Description: *The mint() function declaration uses the 'payable' keyword although no ETH payments are required.*
Risk/Impact: *Any ETH accidentally sent to the contract via this function will be locked in the contract forever. The risk of user confusion resulting in loss of ETH outweighs the small gas savings reaped by including the payable keyword.*
Recommendation: *Remove the 'payable' keyword from the function declaration.*
Resolution: *The team has implemented the above recommendation.*

Finding #5 - Wattbit - Informational (Resolved)

Description: *Several functions are declared public, but are never called internally.*

```
Badge.uri, CBB.withdraw, Sponsor.mint, Sponsor.withdraw
```

Recommendation: *These functions should be declared external for additional gas savings on each call.*
Resolution: *The team has implemented the above recommendation.*

Finding #6 - Badge - Informational (Resolved)

Description: *The* `sponsorNFT` *state variable can never be modified, but is not declared constant.*
Recommendation: *This state variable can be declared constant for additional gas savings on each call.*
Resolution: *The team has removed this state variable entirely; the rest of the code is unaffected by this change.*

CONTRACTS OVERVIEW

- *As the contracts are deployed with Solidity v0.8.10, they are protected from any underflow/overflow issues.*
- *The team worked with us to ensure that the logic is well-structured to avoid reentrancy issues in applicable functions.*
- *The contracts comply with the relevant ERC-1155 and ERC-721 token standards.*

Badge Contract:

- *This contract is used to issue Badge NFTs to users representing ranks.*
- *Anyone can mint themselves one Badge only one time; the Badge type the user will receive is based on their balance of TWL NFTs.*
- *There are 5 types of Badges users may qualify for. The breakdown is as follows:*
  - *1 TWL NFT yields Badge type 1.*
  - *2 TWL NFTs yields Badge type 2.*
  - *3 TWL NFTs yields Badge type 3.*
  - *4 TWL NFTs yields Badge type 4.*
  - *5 and up TWL NFTs yields Badge type 5.*
- *This contract contains a single burn() function that is only accessible via the CBB contract; the purpose of this function is to burn the user's existing Badge NFT and mint the user a new Badge NFT based on their balance of TWL NFTs.*
- *Badges cannot be transferred to any other address at any time.*
- *The owner can set the TWL NFT contract address to any address at any time.*
- *The owner can set the base URI value at any time.*
- *This contract complies with the ERC-1155 multi token standard.*

CBB Contract:

- *This contract is used to facilitate a presale and a public sale for TBI War Lords NFTs.*
- *The maximum supply of TWL NFTs is 10,000.*
- *While the presale is active, whitelisted users can each mint a maximum of 10 NFTs until the maximum supply has been reached; the user must provide a signature that can be used to recover the platform's signature verifier address to ensure that the user is indeed whitelisted.*
- *The owner should exercise caution when granting users access to the presale and ensure that no more than 1,000 users are eligible as there is no total supply check within the mintPresale() function. As such, it is possible that the maximum supply limit may be exceeded.*
- *Each NFT costs 0.05 ETH to mint during the presale.*
- *While the public sale is active, users can mint up to 20 NFTs per transaction until the maximum supply has been reached.*
- *Each NFT initially costs 0.095 ETH to mint during the public sale, but this cost can be changed by the project team at any time.*
- *Users should exercise caution and ensure that the correct amount of ETH is supplied, as the contract will not return any excess ETH to the user.*
- *The owner can mint up to 20 NFTs per transaction until either the maximum reserve supply of 1,000 NFTs has been reached, or the maximum total supply has been reached.*
- *In the event that the address receiving a TWL NFT is a contract, the contract must have implemented the onERC721Received() function in order to successfully receive the NFT.*
- *Upon receiving a TWL NFT, the contract calls the burn() function on the Badge contract in order to ensure the user has the correct Badge type at all times.*
- *The owner can toggle the presale and the public sale at any time.*
- *The owner can set the price of the TWL NFT during the public sale to any value at any time.*
- *The owner can set the base URI value after deployment, but can disable this functionality at any time; once this functionality is disabled, it cannot be enabled again.*
- *The owner can set the unrevealed URI to any value at any time; this value is used as a temporary URI before the true base URI has been revealed.*

> - *The owner can set the Badge contract address to any address at any time; users will not be able to transfer their TWL NFTs to any other address unless the Badge contract address is set.*
> - *The owner can set the signature verifier address to any address at any time.*
> - *The owner can withdraw all the ETH from the contract at any time.*
> - *This contract complies with the ERC-721 standard.*
>
> *Sponsor Contract:*
> - *This contract is used to facilitate the sale of Sponsor NFTs.*
> - *The maximum supply of Sponsor NFTs is 10,000.*
> - *Only users who do not own a Badge NFT are able to mint Sponsor NFTs; however, users can mint a Badge NFT after minting themselves Sponsor NFTs.*
> - *Eligible users can mint up to 20 NFTs per transaction until the maximum supply has been reached.*
> - *Each NFT initially costs 0.085 ETH, but this cost can be changed by the project team at any time.*
> - *Users should exercise caution and ensure that the correct amount of ETH is supplied, as the contract will not return any excess ETH to the user.*
> - *The user can supply an "agent address" during minting, which may be eligible to receive a portion of the amount paid as commission based on their Badge type. The commission schedule is as follows:*
>   - *Badge type 1 yields 10% commission.*
>   - *Badge type 2 yields 20% commission.*
>   - *Badge type 2 yields 30% commission.*
>   - *Badge type 2 yields 40% commission.*
>   - *Badge type 5 yields 50% commission.*
> - *The owner can mint up to 20 NFTs per transaction until either the maximum reserve supply of 100 NFTs has been reached, or the maximum total supply has been reached.*
> - *In the event that the address receiving a Sponsor NFT is a contract, the contract must have implemented the onERC721Received() function in order to successfully receive the NFT.*
> - *The owner can set the price of the Sponsor NFT during to any value at any time.*
> - *The owner can set the base URI value after deployment, but can disable this functionality at any time; once this functionality is disabled, it cannot be enabled again.*
> - *The owner can withdraw all the ETH from the contract at any time.*
> - *This contract complies with the ERC-721 standard.*

AUDIT RESULTS

Vulnerability Category	Notes	Result
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	The owner can set the price of the TWL and Sponsor NFTs at any time.	PASS
Compiler Issues	N/A	PASS
Delegate Call to Untrusted Contract	N/A	PASS
Dependence on Predictable Variables	N/A	PASS
Ether/Token Theft	N/A	PASS
Flash Loans	N/A	PASS
Front Running	N/A	PASS
Improper Events	N/A	PASS
Improper Authorization Scheme	N/A	PASS
Integer Over/Underflow	N/A	PASS
Logical Issues	N/A	PASS
Oracle Issues	N/A	PASS
Outdated Compiler Version	N/A	PASS
Race Conditions	N/A	PASS
Reentrancy	N/A	PASS
Signature Issues	N/A	PASS
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

Badge Contract

INHERITENCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

CBB Contract

INHERITENCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

Sponsor Contract

INHERITENCE CHART

FUNCTION GRAPH

FUNCTIONS OVERVIEW

GO HOME