shieldify

Beetle

SECURITY REVIEW

Date: 29 August 2025

CONTENTS

1. About Shieldify

Positioned as the first hybrid Web3 Security company, Shieldify shakes things up with a unique subscription-based auditing model that entitles the customer to unlimited audits within its duration, as well as top-notch service quality thanks to a disruptive 6-layered security approach. The company works with very well-established researchers in the space and has secured multiple millions in TVL across protocols, also can audit codebases written in Solidity, Vyper, Rust, Cairo, Move and Go.

Learn more about us at shieldify.org.

2. Disclaimer

This security review does not guarantee bulletproof protection against a hack or exploit. Smart contracts are a novel technological feat with many known and unknown risks. The protocol, which this report is intended for, indemnifies Shieldify Security against any responsibility for any misbehavior, bugs, or exploits affecting the audited code during any part of the project's life cycle. It is also pivotal to acknowledge that modifications made to the audited code, including fixes for the issues described in this report, may introduce new problems and necessitate additional auditing.

3. About Beetle

Beetle is the only NFT card game featuring a complex game loop with a synergistic throughput of mechanics, it stands out due to its engaging pack openings, dynamic autobattles, and a collection goal of acquiring 50/50 of the full card set. Lore is not central but serves to excite players about the Beetles, enhancing immersion without overshadowing gameplay.

Battle Structure

- **Party Construction and Summoning**: Players build parties of 2-6 Beetles (cards) without traditional deck restrictions (de-emphasis on randomness affecting player choice, reducing player frustration, rather choice causes randomness to occur in game, and randomness in outcome is a function of interacting with different party combinations). Summon costs are dictated by a star's system: 1-4 stars = 0 cost (Bronze tier) 5-6 stars = +1 cost (Silver tier) 7+ stars = +2 cost (Gold tier) Higher-star summons require supporting Beetles of the same element, adding strategic depth. Stars instantly communicate Beetle value to players, influencing build decisions and increasing the player connection to the individual Beetles.
- **Match Format**: PvP focuses on autobattles in formats like 1v1 duels or small tournaments (e.g., 4-8 players), with risk-reward elements (e.g., wagering NFTs or crypto on outcomes for higher rewards). PvE adapts this to exploratory encounters in a wandering world, where parties face AI-controlled objectives or bosses without player elimination.
- **Combat**: Preparation (party construction) and Combat (auto-battles). No complex grid board; instead, a streamlined attack system where Beetles auto-engage based on stats, elements, and synergies. Additional phases could include post-battle rewards or event triggers (e.g., environmental hazards in PvE).
- **Win Conditions**: In PvP, the last party standing wins, with rewards scaling based on performance (e.g., locked cards or choosables). In PvE, completing objectives like defeating world bosses or collecting items.
- **Post-Battle Rewards**: After a PvP battle is won, the winner can choose one card from the loser's party. Both players can lock 1 card before battle (termed "Frozen" or "Protected") that

cannot be chosen. Thus, the winner selects from the remaining 1/5 Beetles. This is an on-chain NFT transaction where the winner gains the NFT, and the loser loses their card, implementing high-stakes risk-reward into the game loop with mechanistic synergy between risk, strategy, and rewards all centered on Beetle cards.

4. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4.1 Impact

- **High** – results in a significant risk for the protocol's overall well-being. Affects all or most users
- **Medium** – results in a non-critical risk for the protocol affects all or only a subset of users, but is still unacceptable
- **Low** – losses will be limited but bearable, and covers vectors similar to griefing attacks that can be easily repaired

4.2 Likelihood

- **High** – almost certain to happen and highly lucrative for execution by malicious actors
- **Medium** – still relatively likely, although only conditionally possible
- **Low** – requires a unique set of circumstances and poses non-lucrative cost-of-execution to rewards ratio for the actor

5. Security Review Summary

The security review lasted 4 days, with a total of 64 hours dedicated by 2 researchers from the Shieldify team.

Overall, the code is well-written. The audit report flagged one Critical, one Medium and two Low severity issues. The finding with a higher severity is mainly related to the possible denial of an opponent's NFT rewards and total centralisation of seizing user assets.

The Beetle team has done a great job with their test suite and provided exceptional support to the Shieldify researchers.

5.1 Protocol Summary

Project Name	**Beetle**
Repository	BeetleGame
Type of Project	Digital Card Collecting and Battling Game
Audit Timeline	4 days
Review Commit Hash	4ab13a82a1f9b7228d5c0ad3a085518a60459036
Fixes Review Commit Hash	b8ed61ad3126932b2d53a7713e3e168fac5c6aac

5.2 Scope

The following smart contracts were in the scope of the security review:

File	nSLOC
contracts/BeetleBattleGame.sol	198
contracts/BeetleCards.sol	127
Total	**325**

6. Findings Summary

The following issues have been identified, sorted by their severity:

- **Critical** & **High** issues: **1**
- **Medium** issues: **1**
- **Low** issues: **2**

ID	Title	Severity	Status
[C-01]	User Can Deny Opponent NFT Rewards By Marking Safe Post-Battle	Critical	Fixed
[M-01]	Admin Can Arbitrarily Seize User Assets	Medium	Fixed
[L-01]	Incorrect Price Emission In Batch Mint	Low	Fixed
[L-02]	Use `safeMint()` Instead of `mint()`	Low	Fixed

7. Findings

[C-01] User Can Deny Opponent NFT Rewards By Marking Safe Post-Battle

Severity

Critical Risk

Description

The `markBeetleSafe()` function allows a user to designate one of their staked NFTs as the safe beetle at any time. The function only verifies that the caller is the staker of the token, but does not validate the timing or context of the action. As a result, a user can wait until after losing a battle and then call `markBeetleSafe()` on the threatened NFT. Because `claimBeetleFromVictory()` prohibits transferring a safe beetle, this enables a malicious participant to retroactively protect assets that should otherwise be lost.

Location of Affected Code

File: contracts/BeetleBattleGame.sol#L231

```
function markBeetleSafe(uint256 _tokenId) public {
  if (isStaked[_tokenId] != msg.sender) revert InvalidTokenOwner();
  safeBeetle[msg.sender] = _tokenId;

  emit SafeBeetleUpdated(msg.sender, _tokenId);
}
```

Impact

Winners of battles may be unable to claim the opponent's NFTs, as the defeated party can immediately call `markBeetleSafe()` to shield the asset after the outcome is determined. This undermines the fairness of battles, negates the reward mechanism, and weakens trust in the protocol's economic model.

Recommendation

Restrict `markBeetleSafe()` to only be called during the staking process or require a valid signer authorization to confirm the action, preventing post-battle misuse.

Team Response

Fixed.

[M-01] Admin Can Arbitrarily Seize User Assets

Severity

Medium Risk

Description

The `BeetleBattleGame.emergencyUnstake()` function allows an admin to transfer NFTs staked by any user to an arbitrary `_to` address without user consent. While this may be intended for recovery, it effectively grants administrators unilateral control over all user assets.

Location of Affected Code

File: contracts/BeetleBattleGame.sol

Impact

Users' staked NFTs are not fully trustless. If the admin account is compromised or malicious, staked NFTs can be redirected and permanently lost.

Recommendation

Restrict emergency functions to require multi-sig approval.

Team Response

Fixed.

[L-01] Incorrect Price Emission In Batch Mint

Severity

Low Risk

Description

In `_mintCardWithId()`, the CardMinted event emits `msg.value` for each token, even though the value corresponds to the entire batch. This makes event logs misleading, as every card appears to have been purchased for the full batch price.

Location of Affected Code

File: contracts/BeetleCards.sol#L141

```
function _mintCardWithId(address _to, uint256 _tokenId, uint256
    _cardTypeId) internal {
  cardTypeIds[_tokenId] = _cardTypeId;

  _mint(_to, _tokenId);

@> emit CardMinted(_to, _tokenId, _cardTypeId, msg.value);
}
```

Impact

Analytics, off-chain services, or revenue tracking relying on events will misreport actual purchase prices, causing accounting discrepancies.

Recommendation

Divide the total price by the number of tokens and emit the correct per-card price.

Team Response

Fixed.

[L-02] Use `safeMint()` Instead of `mint()`

Severity

Low Risk

Description

The contract was using the `_mint()` function to mint new tokens. OpenZeppelin also defines a function called `_safeMint()` that prevents someone from minting ERC721 to a contract that does not support ERC721 transfers.

The `_safeMint()` flavour of minting causes the recipient of the tokens, if it is a smart contract, to react upon receipt of the tokens.

Location of Affected Code

File: contracts/BeetleCards.sol#L144

```
function _mintCardWithId(address _to, uint256 _tokenId, uint256
    _cardTypeId) internal {
  cardTypeIds[_tokenId] = _cardTypeId;

  _mint(_to, _tokenId);

  emit CardMinted(_to, _tokenId, _cardTypeId, msg.value);
}
```

Impact

NFTs may be irretrievable if minted to contracts that do not support ERC721 transfers.

Recommendation

It is recommended to use `_safeMint()` instead of `_mint()` depending on the contract's requirements.

Team Response

Fixed.

shieldify

Thank you!