our shielding · Your smart contracts, our shielding · Your smart c

shieldify

Kanpai Pandas - Traits

SECURITY REVIEW

Date: 14 May 2024

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 have 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 Kanpai Pandas - Traits

This is a simple NFT (ERC1155) contract that will act as an on-chain version of the traits held by Kanpai Pandas. Currently, holders are able to manage their traits off-chain at ppdex.io. Using these contracts holders will be able to remove a trait from their NFT and transfer it on chain to be sold/traded on marketplaces. Holders will also be able to move the traits back onto their NFTs by burning the tokenized version of the trait and adding it back to the NFTs metadata via our website (ppdex.io).

3.1 Observations

1. The `ERC1155PandaTraits.sol` contract extends `ERC1155UpgradeableBurnable.sol` and the minting can only be performed if granted access.
2. The `BackendMinter.sol` contract uses EIP712 to allow the minting of tokens on `ERC1155PandaTraits.sol` only with approval from the backend.

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 1 day with a total of 24 hours dedicated to the audit by the core Shieldify team.

Overall, the code is well-written. The audit report contributed by identifying two low-severity issues, where potential attackers could exploit the implementation and where security best practices for access control were not followed.

5.1 Protocol Summary

Project Name	**Kanpai Pandas - Traits**
Repository	TraitsAsNFTS
Type of Project	ERC1155 Collection
Audit Timeline	1 day
Review Commit Hash	0db375b63b73572b4071a0fc5f7a79a6edb6adf4
Fixes Review Commit Hash	e10a2101e8a71d72a8ffbab73efcb4dee2337605

5.2 Scope

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

File	nSLOC
contracts/BackendMinter.sol	95
contracts/ERC1155PandaTraits.sol	109
Total	**204**

6. Findings Summary

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

- **Critical** and **High** issues: **0**
- **Medium** issues: **0**
- **Low** issues: **2**

ID	Title	Severity	Status
[L-01]	Attacker Can Initialize the Implementation	Low	Fixed
[L-02]	Access Control Does Not Follow Security Best Practices	Low	Fixed

7. Findings

[L-01] Attacker Can Initialize the Implementation

Severity

Low Risk

Description

The contracts are upgradable, inheriting from the Initializable contract. However, the current implementations are missing the `_disableInitializers()` function call in the constructors. Thus, an attacker can initialize the implementation. Usually, the initialized implementation has no direct impact on the proxy itself, however, it can be exploited in a phishing attack. In rare cases, the implementation might be mutable and may have an impact on the proxy.

Location of Affected Code

File: contracts/ERC1155PandaTraits.sol

Recommendation

It is recommended to call `_disableInitializers()` within the contract's constructor to prevent the implementation from being initialized:

```
+ constructor() {
+     _disableInitializers();
+ }
```

Team Response

Fixed as suggested.

[L-02] Access Control Does Not Follow Security Best Practices

Severity

Low Risk

Description

Both `BackendMinter.sol` and `ERC1155PandaTraits.sol` contracts inherit from OpenZeppelin's `AccessControl` and `AccessControlUpgradeable` libraries. However, they do not follow some security best practices, for example, the DEFAULT_ADMIN_ROLE is also its own admin, meaning it has permission to grant and revoke this role.

Location of Affected Code

File: contracts/BackendMinter.sol

File: contracts/ERC1155PandaTraits.sol

Recommendation

Consider following security best practices and OpenZeppelin's recommendations, and use the `AccessControlDefaultAdminRules` extension to enforce additional security measures over this role.

Team Response

Fixed as suggested.

our shielding · Your smart contracts, our shielding · Your smart c

shieldify

Thank you!