Gods Unchained CARD Token Audit

SMCAuditors

https://smcauditors.com / info@smcauditors.com

2nd of July 2020

This document is the audit of God's Unchained CARD token performed by SMCAuditors.

1. Executive Summary

This report was written to provide a security audit for God's Unchained Trading Card Game ERC721CARD token. Gods Unchained (https://godsunchained.com/) is a trading card game powered by *Immutable* on Ethereum Blockchain where ownership of each card is guaranteed by CARD token smart contract. SMCAuditors conducted the audit to find security vulnerabilities. The contract is deployed on Ethereum mainnet as stated with the link below.

- https://etherscan.io/address/0x0e3a2a1f2146d86a604adc220b4967a898d7fe07

2. Audit Findings

BatchWrapper.sol

Medium Level Findings

1. Out of bounds enum inputs can trigger invalid opcode. When used as an argument in the following functions *_validateProtos* and *copyNextBatch* an out-of-range enum value triggers an INVALID opcode. This is an assert-style exception that burns all of the gas in the transactions. In general, this type of exception should never happen in production code. The REVERT opcode provides the same assurances with the option of including an additional error message if used with require(). The REVERT opcode does not burn all remaining gas sent with the transaction. Consider adding explicit validation for any input with an enum type in non-admin functions. While this needs to be balanced against the effort of refactoring the full code base, heavy use of the enum type is likely creating more complexity than it is clarity.
2. State access in event emitted after *cardsminted* call. In *mintCards*() function, state access occurs after an external transfer call to an arbitrary address. Accessing state after an untrusted external call brings potential risk for reentrancy, particularly now that transfer is able to trigger code execution with the updated storage opcode costs. Consider accessing the state before the call.

Low Level Findings

1. Duplicate values possible in series array. The function *updateProtos()* will accept duplicate values for the Series[] array. Correct trait initialization relies on external validation.
2. *updateprotos()* function does not check for null addresses. It's possible for a sysAdmin to accidentally or maliciously set the address of a new contract into address(0) when calling the *updateProtos*(). Consider adding null validation to the *updateProtos()* function.
3. *burnAll*() and *isTradable*() functions are not used internally and can safely be changed from public to external.

3. Conclusion

Our identified issues are sent to *Immutable* for consideration.

4. Disclaimer

This report is not advice on investment, nor does it guarantee adequacy of a business model and/or a bug-free code.