CFG NINJA AUDITS

Security Assessment

ApeWars NFT

September 10, 2022

Table of Contents

Audit Summary

This report has been prepared for ApeWars NFT on the Binance Smart Chain network. CFGNINJA provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.

Project Overview

Token Summary

Parameter	Result
Address	0xE8C1B9E512992296a5d40b95d5fcc7F10609862b
Name	ApeWars
Token Tracker	ApeWars (ApeWars)
Decimals	0
Supply	10000
Platform	Binance Smart Chain
compiler	v0.8.7+commit.e28d00a7
Contract Name	ApeWars
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/address/0xE8C1B9E512992296a5d40b95d5fcc7F10609862b#code
Payment Tx	0x1fc88d6395bfd8657d9138abba61d01bb725a796c3df62152fb5542c42c51432

Project Overview

Risk Analysis Summary

Parameter	Result
Buy Tax	0%
Sale Tax	0%
Is honeypot?	Clean
Can edit tax?	No
Is anti whale?	No
Is blacklisted?	No
Is whitelisted?	Yes
Holders	Clean
Security Score	95/100
Auditor Score	97/100
Confidence Level	Pass

The following quick summary has been added to the project overview, however there are more details about the audit and their results please read every details.

Main Contract Assessed Contract Name

Name	Contract	Live
ApeWars	0xE8C1B9E512992296a5d40b95d5fcc7F10609862b	Yes

TestNet Contract Assessed Contract Name

Name	Contract	Live
ApeWars	0x362C174FA673EaD85C563F8e340fa5a9D4b1Bf07	Yes

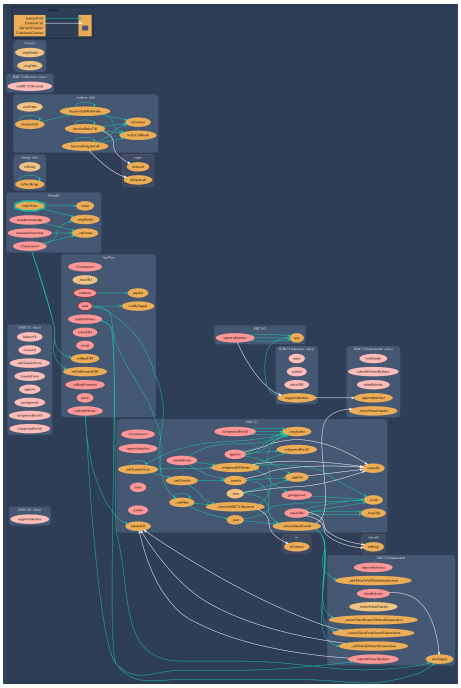
Solidity Code Provided

SolID	File Sha-1	FileName
ApeWars	8bdfa422d572b1d3c0486b46e723dc441794bb18	ApeWars.sol

Call Graph

The contract for ApeWars has the following call graph structure

The Project has a Total Supply of 10000 and has the following inheritance



KYC Information

The Project Onwers of ApeWars has provided KYC Documentation.

KYC Certificated can be found on the Following:
KYC Data

KYC Information Notes:

Auditor Notes: Asked project owner about KYC, Project owner passed KYC IDOPresales

Project Owner Notes:

Smart Contract Vulnerability Checks

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	ApeWars.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	ApeWars.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	ApeWars.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	ApeWars.sol	L: 10 C: 0
SWC-104	Pass	Unchecked Call Return Value.	ApeWars.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	ApeWars.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	ApeWars.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	ApeWars.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	ApeWars.sol	L: 602 C: 9
SWC-109	Pass	Uninitialized Storage Pointer.	ApeWars.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	ApeWars.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	ApeWars.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	ApeWars.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-113	Pass	Multiple calls are executed in the same transaction.	ApeWars.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	ApeWars.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	ApeWars.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	ApeWars.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	ApeWars.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	ApeWars.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	ApeWars.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	ApeWars.sol	L: 420 C: 12,L: 493 C: 28
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	ApeWars.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	ApeWars.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	ApeWars.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	ApeWars.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	ApeWars.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	ApeWars.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	ApeWars.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-128	Pass	DoS With Block Gas Limit.	ApeWars.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	ApeWars.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	ApeWars.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	ApeWars.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	ApeWars.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	ApeWars.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	ApeWars.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	ApeWars.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	ApeWars.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry standard security scanning tool

Smart Contract Vulnerability Details

SWC-103 - Floating Pragma.

CWE-664: Improper Control of a Resource Through its Lifetime.

References:

Description:

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation:

Lock the pragma version and also consider known bugs (https://github.com/ethereum/solidity/releases) for the compiler version that is chosen.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

References:

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

Smart Contract Vulnerability Details

SWC-108 - State Variable Default Visibility

CWE-710: Improper Adherence to Coding Standards

Description:

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

Remediation:

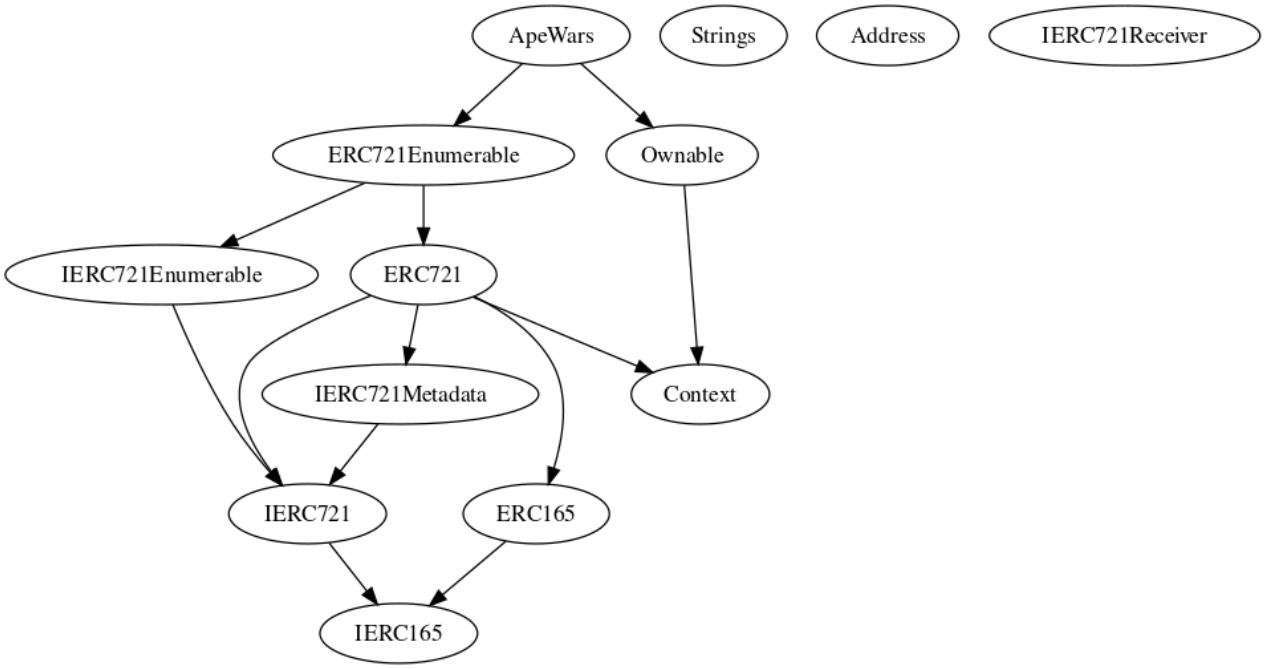
Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

References:

Ethereum Smart Contract Best Practices - Explicitly mark visibility in functions and state variables

Call Graph and Inheritance

The contract for ApeWars has the following call graph structure



Priviliged Functions (onlyOwner)

Function Name	Parameters	Visibility
renounceOwnership	none	external
transferOwnership	none	public
mintForOwner	none	public
reveal	none	external
setNotRevealedURI	none	external
setBaseURI	none	external
setBaseExtension	none	external
pause	none	public
withdraw	none	public

Assessment Results

- Contract is a Rebase
- The following is a NFT Contract.
- No Vulnerabilities have been found on the code.
- Code use and follow the conding best practices.
- Planet Ape has been identified as a SAFU Project and this contract has been audited to ensure investor's safety.

Audit Passed

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/PlanetApeClub	Pass
Instagram	https://www.instagram.com/PlanetApeclub.io/	Pass
Website	http://planetapeclub.io	Pass
Telegram	https://t.me/PlanetApeClub	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes: No other social media

Appendix

Finding Categories

Centralization / Privilege

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

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM 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 howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing 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 mayresult in a vulnerability.

Coding Style

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

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

CFGNINJA has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and CFGNINJA is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will CFGNINJA or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

The assessment services provided by CFGNINJA 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.