SMART CONTRACT AUDIT REPORT

FreeStylers Esport. UNIVERS

APRIL 2023

Approved by

HEXSEC BLOCKCHAIN

HEXSEC
BLOCKCHAIN AUDIT

DISCLAIMER

This report presents a summary of our findings and analysis, based on good industry practices as of the date of this report, concerning potential cybersecurity vulnerabilities and issues within the framework and algorithms used for smart contracts. The report details the specific vulnerabilities and issues identified by our analysis. However, it is important to note that this report is limited in scope and should not be solely relied upon for decision-making purposes. It is crucial to conduct independent investigations and evaluations prior to making any decisions related to the use of smart contracts. Please note that this report is an original work and any resemblance to other reports or sources is unintentional.

INTRODUCTION

Token: FreeStylers Esport. UNIVERS
                FSUNI
Decimals: 18
Supply: 500,000,000 FSEUNI
Type: ERC20
Network: <u>Polygon</u>

Contract:
0x5642cfe183e98673f21e2552b5c59355414970ea

Company and Project:
                FreeStylers Esport Team
www address:
                https://freestylers-esport.com
twitter:
                https://twitter.com/fstylers99
email:
                crypto@freestylers-esport.com

ABOUT OWNER

Company:

        FreeStylers Esport Team

Project Description:

        FreeStylers Esport Team plans to release an ERC20 token that will be used as a means of exchange and payment on their gaming platform.

This token will be used to purchase in-game items and as a reward for participants in esports tournaments and leagues. The token is expected to be introduced to the market in 2023/2024 along with the current technology.

A significant element in the future plans of the project is the introduction of an ERC721 NFT token for the purpose of issuing a collection of standard and rare items.

VULNERABILITY & RISK LEVEL

Risk represents the likelihood that a particular source-threat will exploit a vulnerability, and the impact of that event on the organization or system. Risk level is calculated based on the CVSS version 3.0. Please familiarize yourself with the following designations:"

CVSS = Common Vulnerability Scoring System

Severity levels:

- Critical (CVSS score: 9.0-10.0)
- High (CVSS score: 7.0-8.9)
- Medium (CVSS score: 4.0-6.9)
- Low (CVSS score: 0.2-3.9)
- Informational (CVSS score: 0 – 1.9)

LEVEL VALUE VULNERABILITY RISK (REQUIRED ACTION)

CRITICAL 9 – 10 -RED
A VULNERABILITY THAT CAN
DISRUPT THE CONTRACT
FUNCTIONING IN A NUMBER
OF SCENARIOS, OR CREATES A
RISK THAT THE CONTRACT MAY BE
BROKEN.
IMMEDIATE ACTION TO REDUCE RISK LEVEL.

HIGH 7 – 8.9 - ORANGE
A VULNERABILITY THAT AFFECTS
THE DESIRED OUTCOME WHEN
USING A CONTRACT, OR PROVIDES
THE OPPORTUNITY TO USE A
CONTRACT IN AN UNINTENDED
WAY.
IMPLEMENTATION OF CORRECTIVE ACTIONS AS SOON AS
POSSIBLE.

MEDIUM 4 – 6.9 - YELLOW
A VULNERABILITY THAT COULD
AFFECT THE DESIRED OUTCOME OF
EXECUTING THE CONTRACT IN A
SPECIFIC SCENARIO.
IMPLEMENTATION OF CORRECTIVE ACTIONS IN A CERTAIN
PERIOD.

LOW 2 – 3.9 - GREEN
A VULNERABILITY THAT DOES NOT
HAVE A SIGNIFICANT IMPACT ON
POSSIBLE SCENARIOS FOR THE
USE OF THE CONTRACT AND IS
PROBABLY SUBJECTIVE.
IMPLEMENTATION OF CERTAIN CORRECTIVE ACTIONS OR
ACCEPTING THE
RISK.

INFORMATIONAL 0 – 1.9 - BLUE
A VULNERABILITY THAT HAVE
INFORMATIONAL CHARACTER BUT IS
NOT EFFECTING ANY OF THE
CODE.
AN OBSERVATION THAT DOES NOT DETERM

AUDIT STRATEGY AND TECHNIQUES

DURING THE REVIEW PROCESS, WE DILIGENTLY ASSESSED THE REPOSITORY FOR POTENTIAL SECURITY VULNERABILITIES, CODE QUALITY, AND COMPLIANCE WITH STANDARDS AND BEST PRACTICES. OUR TEAM OF PROFICIENT PENTESTERS AND SMART CONTRACT DEVELOPERS CONDUCTED A THOROUGH LINE-BY-LINE REVIEW, METICULOUSLY DOCUMENTING ANY ISSUES ENCOUNTERED.

METHODOLOGY

DURING THE REVIEW PROCESS, WE DILIGENTLY ASSESSED THE REPOSITORY FOR POTENTIAL SECURITY VULNERABILITIES, CODE QUALITY, AND COMPLIANCE WITH STANDARDS AND BEST PRACTICES. OUR TEAM OF PROFICIENT PENTESTERS AND SMART CONTRACT DEVELOPERS CONDUCTED A THOROUGH LINE-BY-LINE REVIEW, METICULOUSLY DOCUMENTING ANY ISSUES ENCOUNTERED.

METHODOLOGY

1. CODE REVIEW PROCESS: I. WE REVIEWED THE SPECIFICATIONS, SOURCE CODE, AND INSTRUCTIONS PROVIDED TO CHAINSULTING TO ENSURE OUR UNDERSTANDING OF THE SIZE, SCOPE, AND FUNCTIONALITY OF THE SMART CONTRACT. II. WE MANUALLY EXAMINED THE CODE LINE-BY-LINE TO IDENTIFY POTENTIAL VULNERABILITIES. III. WE COMPARED THE CODE TO THE PROVIDED SPECIFICATIONS, SOURCES, AND INSTRUCTIONS TO ENSURE IT MATCHES THE INTENDED FUNCTIONALITY.
2. TESTING AND AUTOMATED ANALYSIS PROCESS: I. WE ANALYZED THE TEST COVERAGE TO DETERMINE IF THE TEST CASES ARE ADEQUATE AND COVERING THE RELEVANT CODE. II. WE PERFORMED SYMBOLIC EXECUTION TO IDENTIFY HOW EACH PART OF THE PROGRAM EXECUTES BASED ON DIFFERENT INPUTS.
3. BEST PRACTICES REVIEW: WE CONDUCTED A REVIEW OF THE SMART CONTRACTS TO IMPROVE EFFICIENCY, EFFECTIVENESS, CLARITY, MAINTAINABILITY, SECURITY, AND CONTROL. THIS WAS BASED ON ESTABLISHED INDUSTRY AND ACADEMIC PRACTICES, RECOMMENDATIONS, AND RESEARCH.
4. SPECIFIC RECOMMENDATIONS: WE PROVIDED SPECIFIC, ITEMIZED, AND ACTIONABLE RECOMMENDATIONS TO HELP YOU TAKE STEPS TO SECURE YOUR SMART CONTRACTS.

TESTED CONTRACT SOURCE FILE

SMART CONTRACTS HAVE BEEN TESTED FOR CORRECTNESS AND VULNERABILITIES.

TESTED FILES:

> FREESTYLERSUNI.SOL
> ERC20.SOL
> IERC.SOL
> OWNABLE.SOL

TESTING TOOLS:

> MYTHX
> TRUFFLE
> SLITHER

RESULT

REFERENCE: HTTPS://GITHUB.COM/CRYTIC/SLITHER/WIKI/DETECTOR-DOCUMENTATION#LOCAL-VARIABLE-SHADOWING
INFO:DETECTORS:

DIFFERENT VERSIONS OF SOLIDITY ARE USED:
 - VERSION USED: ['^0.8.0', '^0.8.10']
 - ^0.8.0 (CONTEXT.SOL#4)
 - ^0.8.0 (ERC20.SOL#4)
 - ^0.8.0 (IERC20.SOL#4)
 - ^0.8.0 (IERC20METADATA.SOL#4)
 - ^0.8.0 (OWNABLE.SOL#4)
 - ^0.8.10 (FREESTYLERSUNI.SOL#3)

NFO:DETECTORS:
CONTEXT._MSGDATA() (CONTEXT.SOL#21-23) IS NEVER USED AND SHOULD BE REMOVED
ERC20._BURN(ADDRESS,UINT256) (ERC20.SOL#280-295) IS NEVER USED AND SHOULD BE REMOVED
REFERENCE: HTTPS://GITHUB.COM/CRYTIC/SLITHER/WIKI/DETECTOR-DOCUMENTATION#DEAD-CODE
INFO:DETECTORS:
PRAGMA VERSION^0.8.0 (CONTEXT.SOL#4) ALLOWS OLD VERSIONS
PRAGMA VERSION^0.8.0 (ERC20.SOL#4) ALLOWS OLD VERSIONS
PRAGMA VERSION^0.8.10 (FREESTYLERSUNI.SOL#3) ALLOWS OLD VERSIONS
PRAGMA VERSION^0.8.0 (IERC20.SOL#4) ALLOWS OLD VERSIONS
PRAGMA VERSION^0.8.0 (IERC20METADATA.SOL#4) ALLOWS OLD VERSIONS
PRAGMA VERSION^0.8.0 (OWNABLE.SOL#4) ALLOWS OLD VERSIONS
SOLC-0.8.10 IS NOT RECOMMENDED FOR DEPLOYMENT

Report for FreestylersUni.sol:

https://dashboard.mythx.io/#/console/analyses/ba06e21c-cdc7-4568-9430-a48d6353ea48

(SWC-103) Floating Pragma	Low	Passed
(SWC-104) Unchecked Call	Low	Passed
(SWC-110) Assert Violation	Low	Passed
(SWC-123) Requirement Violation	Low	Passed

| Finished | mythx-cli-0.7.3 | 2023-04-30T17:31:28.440Z |

	Api	v1.12.3	
Hash	76ef9b41be62e68dbbef434acbd83214		
	Harvey	0.0.66	
	Maru	0.8.2	
	Mythril	0.22.18	

Report for FreestylersUni.sol:
 Contract locking ether found:
 Contract MintableToken (FreestylersUni.sol#24-48) has payable functions:
  - MintableToken.constructor(string,string,uint256) (FreestylersUni.sol#30-36)
 But does not have a function to withdraw the ether
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#contracts-that-lock-ether
INFO:Detectors:
MintableToken.constructor(string,string,uint256).name (FreestylersUni.sol#31) shadows:
 - ERC20.name() (ERC20.sol#62-64) (function)
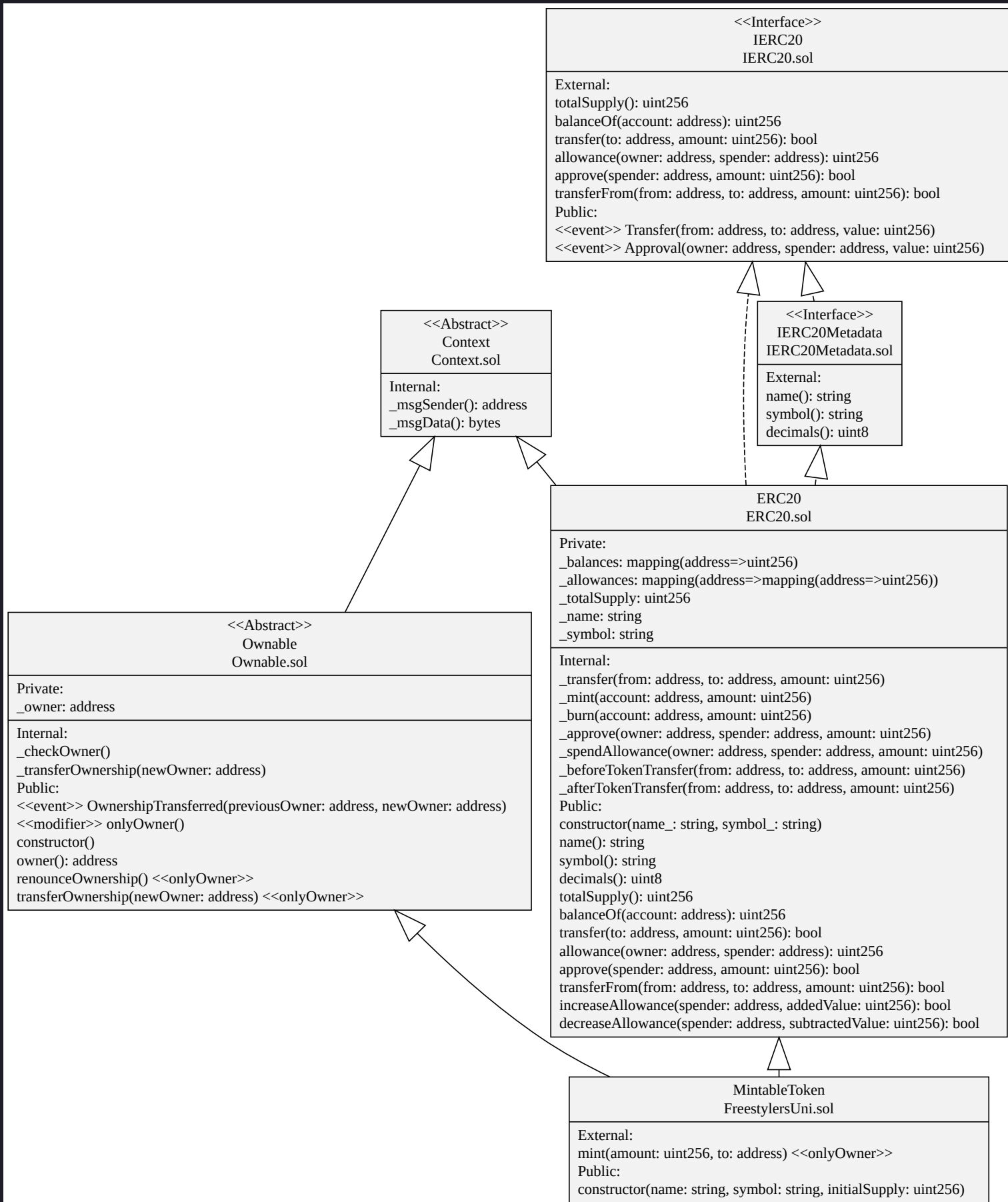 **- IERC20Metadata.name() (IERC20Metadata.sol#17) (function)**
MintableToken.constructor(string,string,uint256).symbol (FreestylersUni.sol#32) shadows:
 - ERC20.symbol() (ERC20.sol#70-72) (function)
 - IERC20Metadata.symbol() (IERC20Metadata.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:

<<Interface>>
IERC20
IERC20.sol

External:
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(to: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(from: address, to: address, amount: uint256): bool
Public:
<<event>> Transfer(from: address, to: address, value: uint256)
<<event>> Approval(owner: address, spender: address, value: uint256)

<<Abstract>>
Context
Context.sol

Internal:
_msgSender(): address
_msgData(): bytes

<<Interface>>
IERC20Metadata
IERC20Metadata.sol

External:
name(): string
symbol(): string
decimals(): uint8

ERC20
ERC20.sol

Private:
_balances: mapping(address=>uint256)
_allowances: mapping(address=>mapping(address=>uint256))
_totalSupply: uint256
_name: string
_symbol: string

Internal:
_transfer(from: address, to: address, amount: uint256)
_mint(account: address, amount: uint256)
_burn(account: address, amount: uint256)
_approve(owner: address, spender: address, amount: uint256)
_spendAllowance(owner: address, spender: address, amount: uint256)
_beforeTokenTransfer(from: address, to: address, amount: uint256)
_afterTokenTransfer(from: address, to: address, amount: uint256)
Public:
constructor(name_: string, symbol_: string)
name(): string
symbol(): string
decimals(): uint8
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(to: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(from: address, to: address, amount: uint256): bool
increaseAllowance(spender: address, addedValue: uint256): bool
decreaseAllowance(spender: address, subtractedValue: uint256): bool

<<Abstract>>
Ownable
Ownable.sol

Private:
_owner: address

Internal:
_checkOwner()
_transferOwnership(newOwner: address)
Public:
<<event>> OwnershipTransferred(previousOwner: address, newOwner: address)
<<modifier>> onlyOwner()
constructor()
owner(): address
renounceOwnership() <<onlyOwner>>
transferOwnership(newOwner: address) <<onlyOwner>>

MintableToken
FreestylersUni.sol

External:
mint(amount: uint256, to: address) <<onlyOwner>>
Public:
constructor(name: string, symbol: string, initialSupply: uint256)

Total Supply provides information about the total token supply:
Yes **Passed**

Balance Of provides account balance of the owner's account
Yes **Passed**

Transfer executes transfers of a specified number of tokens to a specified address
Yes **Passed**

Approve allow a spender to withdraw a set number of tokens from a specified account
Yes **Passed**

Allowance returns a set number of tokens from a spender to the owner
Yes **Passed**

Anti Bot stops some or all bot wallets from interacting with the smart contract
Yes **Passed**

Transfer Ownership executes transfer of contract ownership to a specified wallet
Yes **Passed**

Renounce Ownership executes transfer of contract ownership to a dead address
Yes **Passed**

Auditor's Verdict

The audit team demonstrated a strong understanding of best practices and principles associated with Smart Contract-based projects. Their evaluation of the project was thorough and accurate, and their recommendations were effectively communicated, allowing for easy implementation. At the same time, some areas requiring improvement were identified. Although the project exhibited good practices, there are certain aspects that could be further optimized. The audit team provided specific guidance and recommendations on possible enhancements to the project to further increase its effectiveness.

Smart contract source code has LOW RISK SEVERITY. FreeStylers UNIVERS has PASSED the smart contract audit.

FreeStylers Esport UNIVERS has received an OUTSTANDING SMART CONTRACT SAFETY CERTIFICATION.

HEXSEC
BLOCKCHAIN AUDIT

Notice of Audit Report Verification

Please be informed that the audit report for xxxx can be verified for its authenticity and sources of information directly at audit@hexsec-blockchain or by submitting an inquiry on our official website at www.hexsec-blockchain.com

Please make sure to provide the project name and the audit date in your inquiry.

We understand that transparency and accountability are crucial in ensuring the credibility and reliability of the audit process. As such, we welcome any inquiries regarding our audit reports and will be happy to provide any necessary information to verify the accuracy of the report and the sources of information used.

CYBERSECURITY BLOCKCHAIN DEVELOPERS SYSTEMS
- 1370 BROADWAY, NEW YORK , NY 10018
- SUPPORT@HEXSEC-BLOCKCHAIN.COM