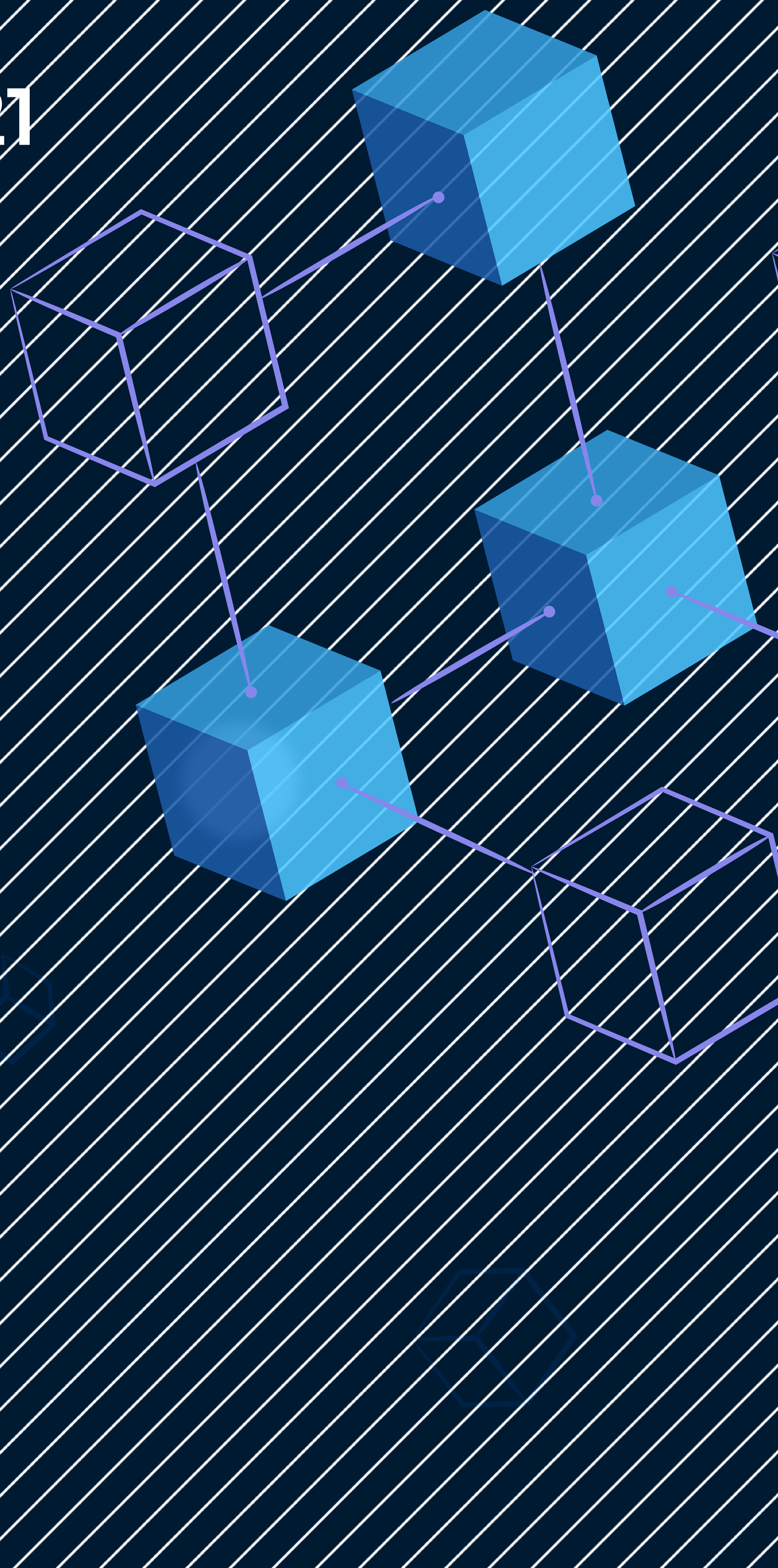
QuillAudits

Audit Report
December, 2021



For

Angels Creed
Angelscreed.com

Contents

Scope of the Audit

The scope of this audit was to analyze and document the AngelsCreed Token Token smart contract codebase for quality, security, and correctness.

Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis
In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis
Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis
Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption
In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit
Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis, Theo.

Issue Categories

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

Risk-level	Description
High	A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.
Medium	The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.
Low	Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.
Informational	These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of issues per severity

Type	High	Medium	Low	Informational
Open	0	0	0	0
Acknowledged	0	0	0	1
Closed	0	0	0	0

Introduction

During the period of **Dec 14, 2021 to Dec 16, 2021** - QuillAudits Team performed a security audit for AngelsCreed Token smart contracts.

The code for the audit was taken from following the official link:
https://github.com/angelscreed-dev/contracts

V	Date	Commit ID
1	15/12/2021	a8ce4fc0f76bb69458fc8dc4e13b7cf81414f972

Issues Found – Code Review / Manual Testing

High severity issues

No issues were found.

Medium severity issues

No issues were found.

Low severity issues

No issues were found.

Informational issues

1. **Missing Events for Significant Transactions**

   Description
   The critical settings are completely devoid of event definitions or emissions. This makes it very difficult for users or other interested parties to track important changes that take place in the system.

   Thus, the missing event makes it difficult to track off-chain liquidity fee changes. An event should be emitted for a critical variable _cap in the burn() function.

   Remediation
   We recommend emitting an event to log the update of the above variable.

   Status: Acknowledged

Functional test

Evn: BSC Tesnet
Contract: 0xA142dbFcE36aCFdDf24dbCC4b45d199130167c0C

Function name	Input	Output	TX	Status
mint (for the owner)	10000000	true	0x030cc0cf76e8e5e80dea7d7c9bbd2a47538cb44378aa2a3abc0f5ba65af1b5ed	Passed
mint (for the owner)	100000000000000000000000000	execution reverted: BEP20Capped: cap exceeded	N/A	Passed
transfer	"0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","10000000"	true	0xc53c560ad6db5ed41ccc672ec6c3dd2d7064078ba4f5becd56a773ff4f4005d0	Passed
balanceOf	0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC	10000000	N/A	Passed
balanceOf	0x1dd64394E29c5988f04A8E074D0DBACd4D614729	0	N/A	Passed
mint	"0x153b057d5d7262dC92099B59c975255ecE66784F","200000000"	true	0xf189767f46386f375b57b03c20236f9471ca7f492b96904640d810fd972429ba	Passed
balanceOf	0x153b057d5d7262dC92099B59c975255ecE66784F	200000000	N/A	Passed
approve	"0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05","200000000"	true	0x566f44cbbb4e7a8a362b0a2a7e2e3bb540e5f8da5c2bbc6c0c02332b89b366c0	Passed
allowance	"0x153b057d5d7262dC92099B59c975255ecE66784F","0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05"	200000000	N/A	Passed

decreaseAllowance	"0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05","200000000"	true	0xd5c713cc88d11479833c252486578418141c56dfdbdbb303a5e620539d3ab66	Passed
allowance	"0x153b057d5d7262dC92099B59c975255ecE66784F","0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05"	0	N/A	Passed
increaseAllowance	"0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05","200000000"	true	0xa3d7f827731b92a49e9cf29a1c2f285086d04c1137b97234b4f56c4f71e7023a	Passed
allowance	"0x153b057d5d7262dC92099B59c975255ecE66784F","0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05"	200000000	N/A	Passed
transferFrom	"0x153b057d5d7262dC92099B59c975255ecE66784F","0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05", 100000000	true	0xa70ad44c91d9e3ea8dff81d5d604b0a391da29db17ace2095426816f442252bd	Passed
allowance	"0x153b057d5d7262dC92099B59c975255ecE66784F","0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05"	100000000	N/A	Passed
balanceOf	0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05	100000000	N/A	Passed
burn (from 0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC)	10000000	true	0x374937ab3fca2f33a1b3da87eb96a94d5029cd2d7dce6b481a6b3b387c7a6bf5	Passed
transferOwnership	0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC	true	0xb669b6f39faf8ad794d7902f01416cd6807ef8bf06863dec314898a0f8262748	Passed

Automated Tests

Slither

```
INFO:Detectors:
BEP20.constructor(string,string).name (angelsCreedToken.sol#410) shadows:
        - BEP20.name() (angelsCreedToken.sol#426-428) (function)
        - IBEP20.name() (angelsCreedToken.sol#112) (function)
BEP20.constructor(string,string).symbol (angelsCreedToken.sol#410) shadows:
        - BEP20.symbol() (angelsCreedToken.sol#440-442) (function)
        - IBEP20.symbol() (angelsCreedToken.sol#107) (function)
BEP20.allowance(address,address).owner (angelsCreedToken.sol#478) shadows:
        - Ownable.owner() (angelsCreedToken.sol#62-64) (function)
BEP20._approve(address,address,uint256).owner (angelsCreedToken.sol#695) shadows:
        - Ownable.owner() (angelsCreedToken.sol#62-64) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
AngelsCreedToken.burn(uint256) (angelsCreedToken.sol#765-768) should emit an event for:
        - _cap = _cap - (amount) (angelsCreedToken.sol#766)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Ownable.constructor().msgSender (angelsCreedToken.sol#54) lacks a zero-check on :
                - _owner = msgSender (angelsCreedToken.sol#55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Address.isContract(address) (angelsCreedToken.sol#209-220) uses assembly
        - INLINE ASM (angelsCreedToken.sol#216-218)
Address._functionCallWithValue(address,bytes,uint256,string) (angelsCreedToken.sol#332-360) uses assembly
        - INLINE ASM (angelsCreedToken.sol#352-355)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address._functionCallWithValue(address,bytes,uint256,string) (angelsCreedToken.sol#332-360) is never used and should be removed
Address.functionCall(address,bytes) (angelsCreedToken.sol#267-272) is never used and should be removed
Address.functionCall(address,bytes,string) (angelsCreedToken.sol#280-286) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (angelsCreedToken.sol#299-311) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (angelsCreedToken.sol#319-330) is never used and should be removed
Address.isContract(address) (angelsCreedToken.sol#209-220) is never used and should be removed
Address.sendValue(address,uint256) (angelsCreedToken.sol#238-247) is never used and should be removed
BEP20._burnFrom(address,uint256) (angelsCreedToken.sol#712-719) is never used and should be removed
Context._msgData() (angelsCreedToken.sol#23-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.9 (angelsCreedToken.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

```
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (angelsCreedToken.sol#238-247):
        - (success) = recipient.call{value: amount}() (angelsCreedToken.sol#242)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (angelsCreedToken.sol#332-360):
        - (success,returndata) = target.call{value: weiValue}(data) (angelsCreedToken.sol#341-343)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter AngelsCreedToken.mint(address,uint256)._to (angelsCreedToken.sol#756) is not in mixedCase
Parameter AngelsCreedToken.mint(address,uint256)._amount (angelsCreedToken.sol#756) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (angelsCreedToken.sol#24)" inContext (angelsCreedToken.sol#14-27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
AngelsCreedToken.slitherConstructorVariables() (angelsCreedToken.sol#728-769) uses literals with too many digits:
        - _cap = 1000000000e18 (angelsCreedToken.sol#729)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (angelsCreedToken.sol#78-80)
name() should be declared external:
        - BEP20.name() (angelsCreedToken.sol#426-428)
decimals() should be declared external:
        - BEP20.decimals() (angelsCreedToken.sol#433-435)
symbol() should be declared external:
        - BEP20.symbol() (angelsCreedToken.sol#440-442)
balanceOf(address) should be declared external:
        - BEP20.balanceOf(address) (angelsCreedToken.sol#454-456)
transfer(address,uint256) should be declared external:
        - BEP20.transfer(address,uint256) (angelsCreedToken.sol#466-473)
allowance(address,address) should be declared external:
        - BEP20.allowance(address,address) (angelsCreedToken.sol#478-485)
approve(address,uint256) should be declared external:
        - BEP20.approve(address,uint256) (angelsCreedToken.sol#494-501)
transferFrom(address,address,uint256) should be declared external:
        - BEP20.transferFrom(address,address,uint256) (angelsCreedToken.sol#515-532)
increaseAllowance(address,uint256) should be declared external:
        - BEP20.increaseAllowance(address,uint256) (angelsCreedToken.sol#546-556)
decreaseAllowance(address,uint256) should be declared external:
        - BEP20.decreaseAllowance(address,uint256) (angelsCreedToken.sol#572-587)
mint(uint256) should be declared external:
        - BEP20.mint(uint256) (angelsCreedToken.sol#597-600)
cap() should be declared external:
        - AngelsCreedToken.cap() (angelsCreedToken.sol#731-733)
mint(address,uint256) should be declared external:
        - AngelsCreedToken.mint(address,uint256) (angelsCreedToken.sol#756-758)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

SOLHINT LINTER

```
angelsCreedToken.sol
   1:1   error    Compiler version 0.8.9 does not satisfy the ^0.5.8 semver requirement                          compiler-version
  17:3   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
  17:17  warning  Code contains empty blocks                                                                     no-empty-blocks
  53:3   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
  86:5   warning  Error message for require is too long                                                          reason-string
 243:5   warning  Error message for require is too long                                                          reason-string
 325:5   warning  Error message for require is too long                                                          reason-string
 410:3   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
 523:5   warning  Error message for require is too long                                                          reason-string
 578:5   warning  Error message for require is too long                                                          reason-string
 621:5   warning  Error message for require is too long                                                          reason-string
 622:5   warning  Error message for require is too long                                                          reason-string
 627:5   warning  Error message for require is too long                                                          reason-string
 667:5   warning  Error message for require is too long                                                          reason-string
 672:5   warning  Error message for require is too long                                                          reason-string
 699:5   warning  Error message for require is too long                                                          reason-string
 700:5   warning  Error message for require is too long                                                          reason-string
 725:22  warning  Code contains empty blocks                                                                     no-empty-blocks
```

Results

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the AngelsCreed Token platform. We performed our audit according to the procedure described above.

The audit showed an informational severity issue which has been Acknowledged By The AngelsCreed Team.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the AngelsCreed Token platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the AngelsCreed Token Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Audit Report
December, 2021

For

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com