QuillAudits

Audit Report September, 2021

For

GAME ZILLA THE GAMING LAUNCHPAD

Contents

Contents

Scope of the Audit

The scope of this audit was to analyze and document the GameZilla 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 ERC-20 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
Mythril, Slither, SmartCheck, Surya, Solhint.

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 four 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	1	3
Acknowledged	0	0	0	0
Closed	0	0	0	0

Introduction

During the period of **September 21, 2021, to September 22, 2021** - QuillAudits Team performed a security audit for GameZillaToken smart contract.

The code for the audit was taken from the following official file:
https://bscscan.com/address/0x4178934c6e313a062c5addd66ab0d9b8d858347a#code

Issues Found

High severity issues

No issues were found.

Medium severity issues

No issues were found.

Low level severity issues

1. **Floating pragma**

   pragma solidity ^0.8.6;

   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.

   Reference
   https://consensys.github.io/smart-contract-best-practices/recommendations/#lock-pragmas-to-specific-compiler-version
   https://swcregistry.io/docs/SWC-103

   Status: Open

Informational

2. A BEP20 token must implement the interface IBEP20 in IBEP20.sol

3. balances and allowed variables should be declared as private

4. balanceOf() should ideally be 'external view returns' instead of 'public view returns'.

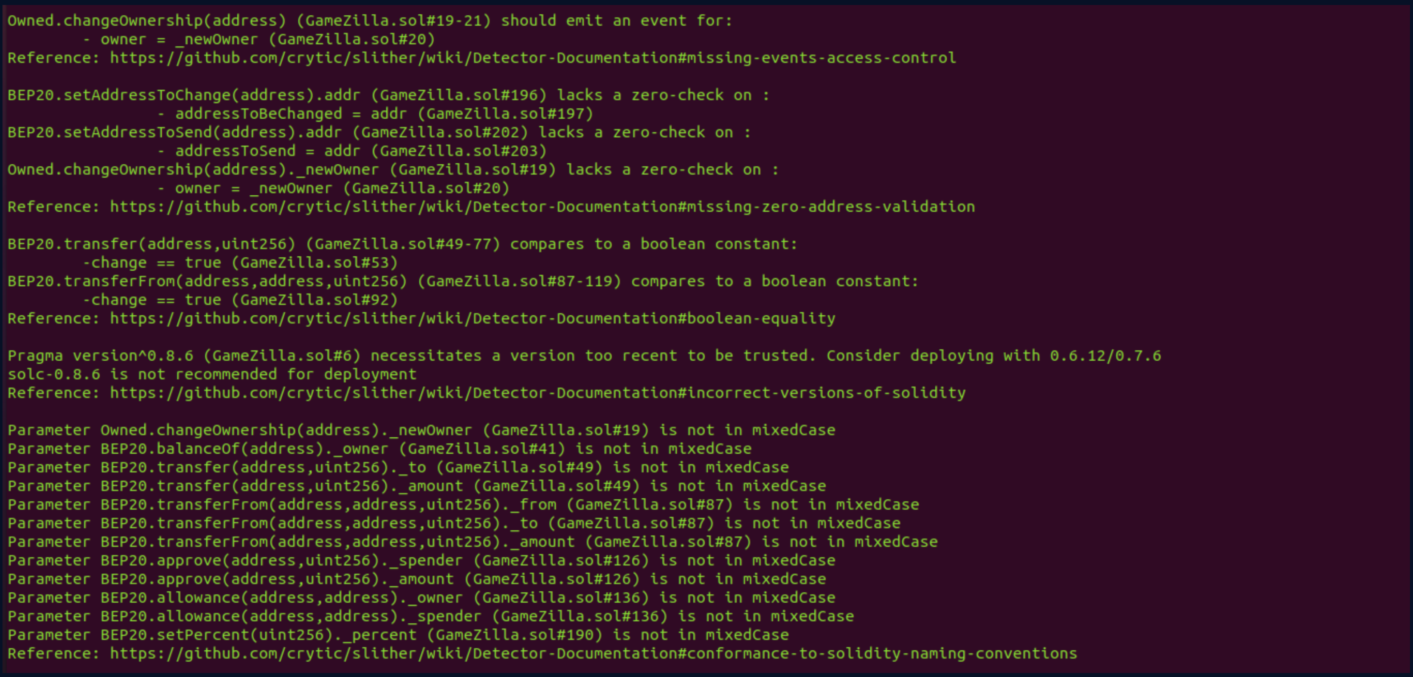
Functional Tests

Function Names	Testing results
approve	Passed
burn	Passed
changeOwnership	Passed
mint	Passed
setAddressToChange	Passed
setAddressToSend	Passed
setPercent	Passed
transferFrom	Passed
transfer	Passed

Automated Testing

Slither

No major issues were found. There are just some Low and Informational errors that were reported by the tool.



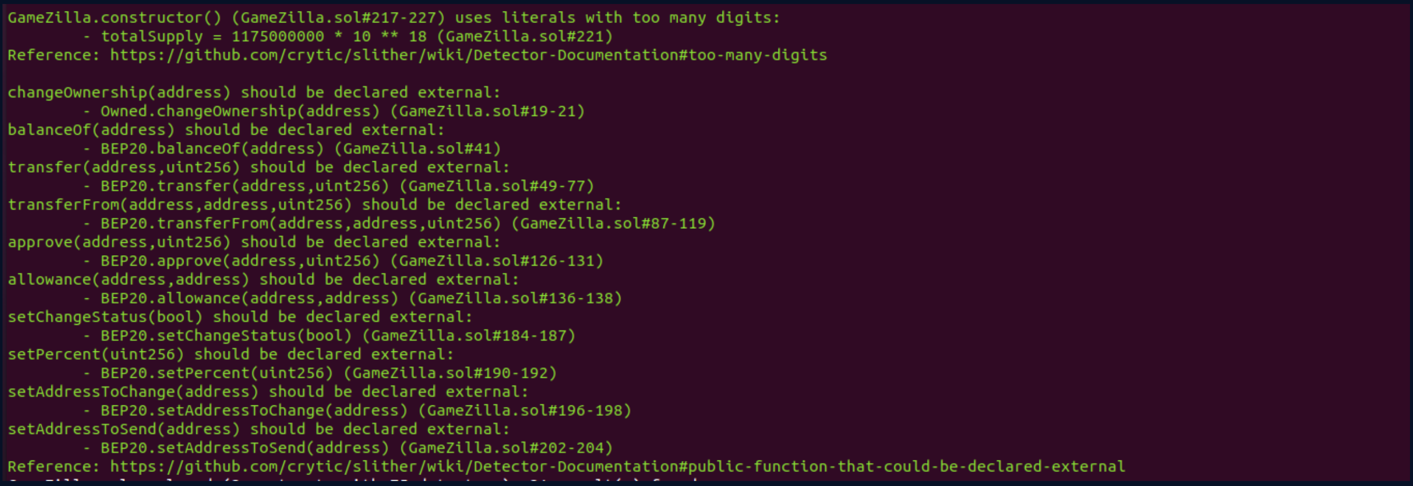
```
Owned.changeOwnership(address) (GameZilla.sol#19-21) should emit an event for:
        - owner = _newOwner (GameZilla.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control

BEP20.setAddressToChange(address).addr (GameZilla.sol#196) lacks a zero-check on :
                - addressToBeChanged = addr (GameZilla.sol#197)
BEP20.setAddressToSend(address).addr (GameZilla.sol#202) lacks a zero-check on :
                - addressToSend = addr (GameZilla.sol#203)
Owned.changeOwnership(address)._newOwner (GameZilla.sol#19) lacks a zero-check on :
                - owner = _newOwner (GameZilla.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

BEP20.transfer(address,uint256) (GameZilla.sol#49-77) compares to a boolean constant:
        -change == true (GameZilla.sol#53)
BEP20.transferFrom(address,address,uint256) (GameZilla.sol#87-119) compares to a boolean constant:
        -change == true (GameZilla.sol#92)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Pragma version^0.8.6 (GameZilla.sol#6) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter Owned.changeOwnership(address)._newOwner (GameZilla.sol#19) is not in mixedCase
Parameter BEP20.balanceOf(address)._owner (GameZilla.sol#41) is not in mixedCase
Parameter BEP20.transfer(address,uint256)._to (GameZilla.sol#49) is not in mixedCase
Parameter BEP20.transfer(address,uint256)._amount (GameZilla.sol#49) is not in mixedCase
Parameter BEP20.transferFrom(address,address,uint256)._from (GameZilla.sol#87) is not in mixedCase
Parameter BEP20.transferFrom(address,address,uint256)._to (GameZilla.sol#87) is not in mixedCase
Parameter BEP20.transferFrom(address,address,uint256)._amount (GameZilla.sol#87) is not in mixedCase
Parameter BEP20.approve(address,uint256)._spender (GameZilla.sol#126) is not in mixedCase
Parameter BEP20.approve(address,uint256)._amount (GameZilla.sol#126) is not in mixedCase
Parameter BEP20.allowance(address,address)._owner (GameZilla.sol#136) is not in mixedCase
Parameter BEP20.allowance(address,address)._spender (GameZilla.sol#136) is not in mixedCase
Parameter BEP20.setPercent(uint256)._percent (GameZilla.sol#190) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

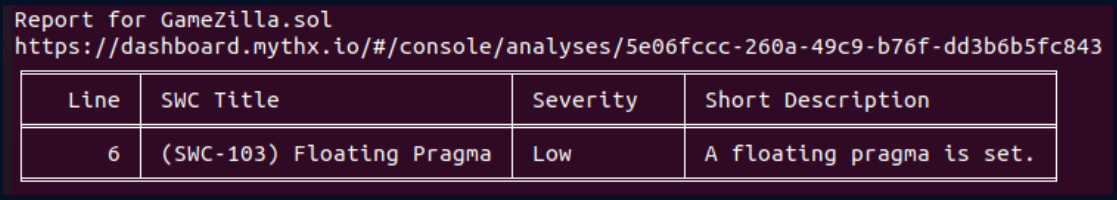


```
GameZilla.constructor() (GameZilla.sol#217-227) uses literals with too many digits:
        - totalSupply = 1175000000 * 10 ** 18 (GameZilla.sol#221)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

changeOwnership(address) should be declared external:
        - Owned.changeOwnership(address) (GameZilla.sol#19-21)
balanceOf(address) should be declared external:
        - BEP20.balanceOf(address) (GameZilla.sol#41)
transfer(address,uint256) should be declared external:
        - BEP20.transfer(address,uint256) (GameZilla.sol#49-77)
transferFrom(address,address,uint256) should be declared external:
        - BEP20.transferFrom(address,address,uint256) (GameZilla.sol#87-119)
approve(address,uint256) should be declared external:
        - BEP20.approve(address,uint256) (GameZilla.sol#126-131)
allowance(address,address) should be declared external:
        - BEP20.allowance(address,address) (GameZilla.sol#136-138)
setChangeStatus(bool) should be declared external:
        - BEP20.setChangeStatus(bool) (GameZilla.sol#184-187)
setPercent(uint256) should be declared external:
        - BEP20.setPercent(uint256) (GameZilla.sol#190-192)
setAddressToChange(address) should be declared external:
        - BEP20.setAddressToChange(address) (GameZilla.sol#196-198)
setAddressToSend(address) should be declared external:
        - BEP20.setAddressToSend(address) (GameZilla.sol#202-204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Mythx

There are just some Low errors that were reported by the tool.

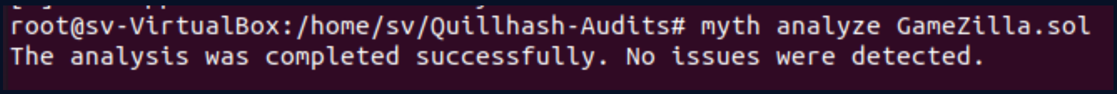


```
Report for GameZilla.sol
https://dashboard.mythx.io/#/console/analyses/5e06fccc-260a-49c9-b76f-dd3b6b5fc843
```

Line	SWC Title	Severity	Short Description
6	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Mythril

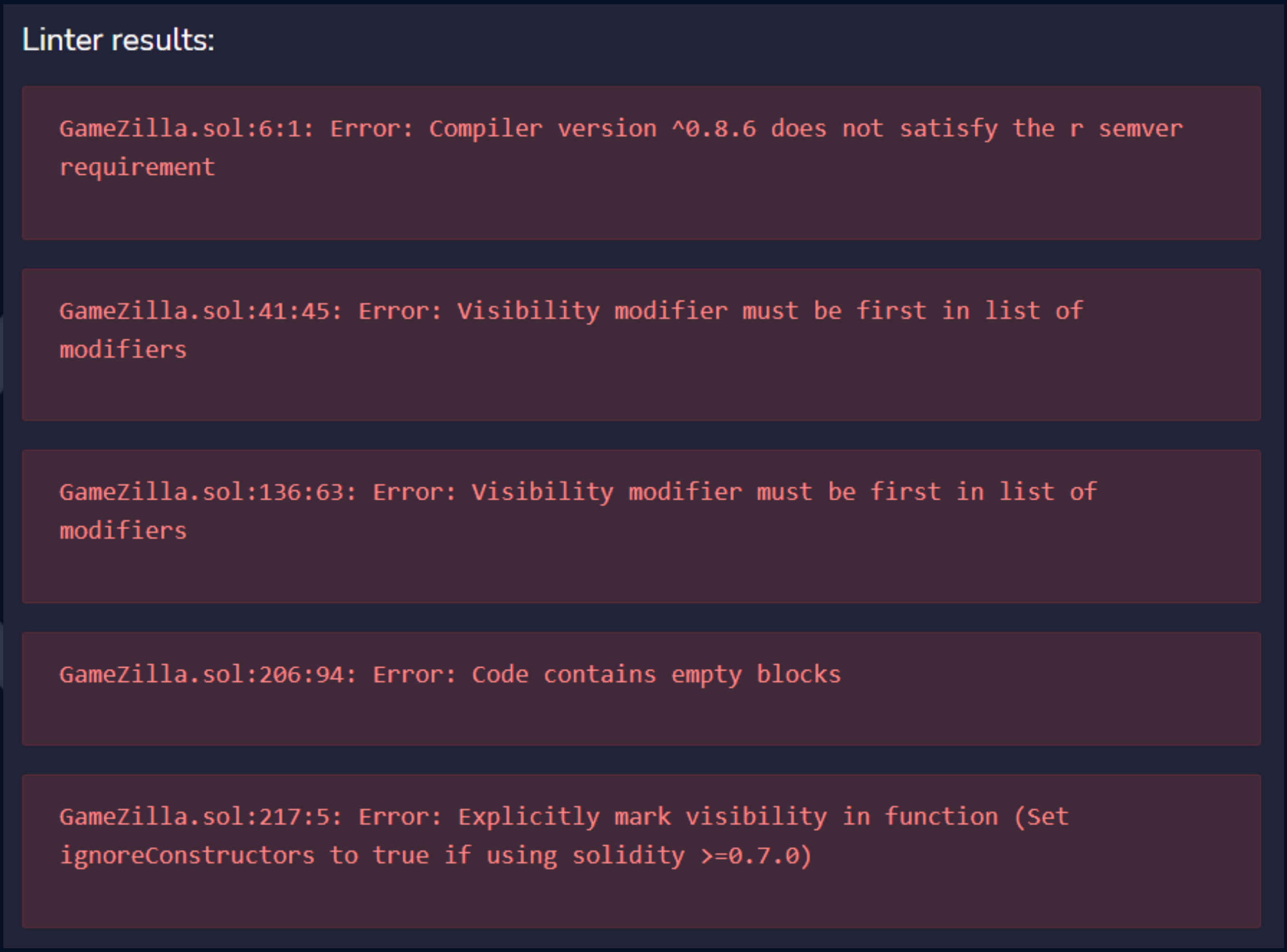
There are just some Low errors that were reported by the tool.



```
root@sv-VirtualBox:/home/sv/Quillhash-Audits# myth analyze GameZilla.sol
The analysis was completed successfully. No issues were detected.
```

Solhint

There are just some basic errors that were reported by the tool.



Linter results:

```
GameZilla.sol:6:1: Error: Compiler version ^0.8.6 does not satisfy the r semver
requirement
```

```
GameZilla.sol:41:45: Error: Visibility modifier must be first in list of
modifiers
```

```
GameZilla.sol:136:63: Error: Visibility modifier must be first in list of
modifiers
```

```
GameZilla.sol:206:94: Error: Code contains empty blocks
```

```
GameZilla.sol:217:5: Error: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)
```

Closing Summary

Overall, in the Initial audit, there is 1 low severity and 3 informational issues associated with the token. It is recommended to fix these before deployment. Overall the optimization issue is neglectable. No instances of Integer Overflow and Underflow, Reentrancy, or any other major vulnerabilities are found in the contract.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the GameZilla Token. 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 GameZilla Token team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Audit Report September, 2021

For

GAME ZILLA THE GAMING LAUNCHPAD

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com