

## **Magic Square**

# SQRVesting Smart Contract Audit Interim Report

Ver. 1.1 May 04, 2024





#### Table of Contents:

Table of Contents	2
Vulnerabilities found by type	
Methodology	
1. SQRVesting.sol	
Verification checksums	11

#### Vulnerabilities found by type:

INFO	1
MEDIUM	0
CRITICAL	0
TOTAL:	1





#### Methodology

During audit process we have analyzed multiple aspects, such as:

- Manual code analysis
- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Locked ether
- Pool Asset Security (backdoors in the underlying ERC-20)
- FA2 compliance (if applicable)
- Logical bugs & code logic issues
- Error handling issues
- General Denial Of Service (DOS)
- Cryptographic errors
- Weak PRNG issues
- Protocol and header parsing errors
- Private data leaks
- Using components with known vulnerabilities
- Unchecked call return method
- Code with no effects
- Unused vars
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Overflows / Underflows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

We have located vulnerabilities listed below.





## 1. SQRVesting.sol

#### Contract methods analysis:

```
INFO
constructor(
address _newOwner,
address _erc20Token,
uint32 _startDate,
uint32 _cliffPeriod,
uint256 _firstUnlockPercent,
uint32 _unlockPeriod,
uint256 _unlockPeriodPercent
)
Consider adding sanity check that _firstUnlockPercent <=
PERCENT_DIVIDER
```

```
getBalance()
Vulnerabilities not detected
Math issues not detected
```



canClaim(address account)
Vulnerabilities not detected
Math issues not detected
calculatePassedPeriod()
Vulnerabilities not detected
Math issues not detected
calculateMaxPeriod()
Vulnerabilities not detected
Math issues not detected

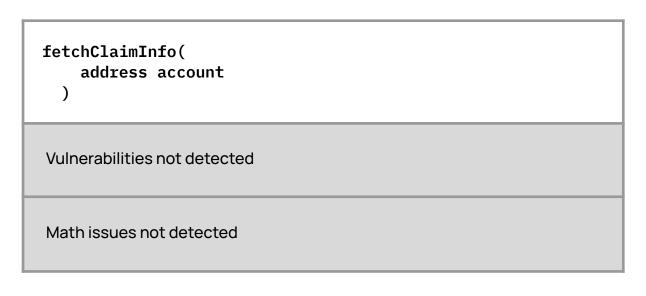


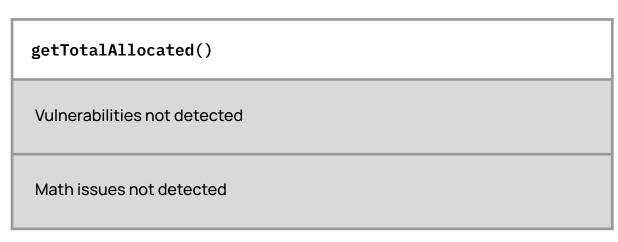
calculateFinishDate()
Vulnerabilities not detected
Math issues not detected
calculateClaimAmount(address account)
Vulnerabilities not detected
Math issues not detected
getAllocationCount()
Vulnerabilities not detected
Math issues not detected



isAllocationFinished(address account)
Vulnerabilities not detected
Math issues not detected
calculateNextClaimAt(address account)
Vulnerabilities not detected
Math issues not detected
calculateRemainAmount(address wallet)
Vulnerabilities not detected
Math issues not detected







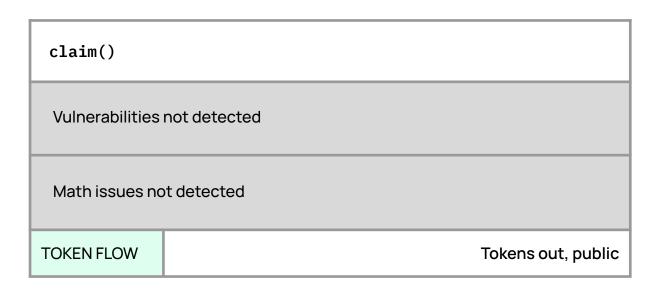
 ${\tt calculatedRequiredAmount()}$ Vulnerabilities not detected Math issues not detected



calculateExcessAmount()
Vulnerabilities not detected
Math issues not detected
_setAllocation(address account, uint256 amount)
Vulnerabilities not detected
Math issues not detected
setAllocation(address account, uint256 amount)
Vulnerabilities not detected
Math issues not detected



```
setAllocations(
    address[] calldata recepients,
    uint256[] calldata amounts
  )
Vulnerabilities not detected
Math issues not detected
```



```
withdrawExcessAmount()
 Vulnerabilities not detected
 Math issues not detected
TOKEN FLOW
                                                 Tokens out, onlyOwner
```





## **Verification checksums**

Contract name	Bytecode hash (SHA-256)
SQRVesting.sol	60ef46c4f1b140571d5e92cf0de2e2af1da6d fd0114fd28e48af4468e1ed5358