OBJECT COMPUTING

Object Computing, Inc.
12140 Woodcrest Executive Dr.
Suite 300
St. Louis, MO 63141
+1 314.579.0066

Vesting Contract Recommendations

- ***Potential* Reentrancy: Minor**
  - *Line #108* ***may*** *need to go after #123 as an additional reentrancy prevention (in addition to the* ***nonReentrant*** *modifier)*
- **Using timestamp for comparisons: Minor** Lines #86-89, #132-140, #152-164
  - It is recommended to avoid using **block.timestamp (lines #111, #134)** and instead use **block.number**, as miners have the ability to adjust timestamps slightly.
- **The onlyBeneficiary modifier is not called:** Unless the usage was overlooked, it should be removed: **Remark**
- **Usage of Public keyword in constructor (line 40) is redundant (Visibility for constructor is ignored): Remark**
- **Move nonReentrant modifier to top of file with other modifiers: Remark**
- **Emit should go at end of function (standard practice) (line #160 after #162)*:*** **Remark**
- **Deploy to testnet and ensure contract is working as expected: Remark**

Slither Results

No	Message	Line #'s	Severity	Reference	Recommendation
1	contains a tautology or contradiction	#173-197 - scheduleIndex >= 0 (contracts/AlienWorldsVesting.sol#184)	Remark	https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction	Change line #184 to `scheduleIndex > 0`
2	lacks a zero-check on	- treasuryVestingAdmin = treasuryVestingAdministrator (contracts/AlienWorldsVesting.sol#99)	Remark	https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation	Require that `treasuryVestingAdministrator` is not the zero address
No	Message	Line #'s	Severity	Reference	Recommendation
3	uses timestamp for comparisons	#109-159	Minor	https://github.com/crytic/slither/wiki/Detector-	No change recommended, as instructed by dallas johnson to keep the rest of

				Documentation#block-timestamp	business logic the same.
3.a	Dangerous comparisons	#132	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
4	uses timestamp for comparisons	#173-197	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
4.a	Dangerous comparisons	#178 #187	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
5	uses timestamp for comparisons	#203-210	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
5.a	Dangerous comparisons	#206	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
6	uses timestamp for comparisons	#216-223	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.
6.a	Dangerous comparisons	#219	Minor	https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp	No change recommended, as instructed by dallas johnson to keep the rest of business logic the same.

Difference Code Review (AlienWorldsVesting.sol)

Line #	Diff	Jack's Comment	Dallas Response
19-20	added new public address variable called `treasuryVestingAdmin`	`treasureyVestingAdmin` could be made private. If the outside world needs to read it, you could create a getTreasureyVestingAdmin	
23	removing and changing variable names in `BeneficiaryVestingRecord` struct	None	
31-36	changing `BeneficiaryAdded` event arguments	None	
40	New `TreasuryVestingAdminSet` event added	None	
45-49	Renaming `tokenAddress` to `_tokenAddress`	None	
45-49	Adding `_treasuryVestingAdmin` to the constructor and only setting the contracts `treasuryVestingAdmin` if the address is not zero	None	
69	Changing the "Beneficiary doesn't have any vesting record" to check `beneficiary` address instead of `vestingStartTime`	None	
88-91	Adding `onlyTreasuryVestingAdmin()` modifier function	None	
93-101	Adding a `setTreasuryVestingAdmin()` function	Add the same require check in the constructor to check that the new address is not zero. But the `onlyOwner` modifier does take care of that for you	
103-113	Changing `addVesting()` arguments	None	
118	Checking that vesting beneficiary is not zero	None	
123-124	Refactoring vesting amount non zero	None	

	check		
Line #	Diff	Jack's Comment	Dallas Response
127-137	Checking vesting dates	Is `++scheduleIndex` supposed to be `scheduleIndex++` ?	This is intended to optimize gas cost. https://www.reddit.com/r/ethdev/comments/tcwspw/i_vs_i_gas_efficiency/i0kebl4/
139	Calculating `amountForVesting` differently	None	
145-149	Insinuating `beneficiaryVestingRecord` with the new parameters	None	
154-156	Emitting `BeneficiaryAdded` event with new parameters	None	
161-167	Adding `getVestingSchedules()` function	None	
178	Refactoring return 0 "short circuit" when vesting not available	None	
182-196	Refactoring get vesting period logic	None	
199-202	Adding function comment	None	
206	Refactoring vesting schedule check	None	
212-215	Adding function comments	None	
219	Refactoring vesting schedule check	None	
225-227	Adding function comment	None	
233	Removing the "Beneficiary does not have vesting to claim" check.	If `claimableAmount` is zero, will this function succeed yet do nothing? Is this intended or will users be frustrated spending gas on a function that does nothing for them?	PR: https://github.com/Alien-Worlds/alienworlds-vesting/commit/cb11f6884b914cd0a35c09a16456b657ed7ac040

No methodology definitively proves the absence of vulnerabilities. Following assessment and remediation, modifications to an application, its platform, network environment, and new threat vectors may result in new application security vulnerabilities.