OVERVIEW

PROJECT SUMMARY

Project	**CryptoYachts**
Platform	N/a
Language	Solidity

AUDIT SUMMARY

Date	02-03-2022
Audit Type	Static Analysis, Manual Review
Audit Result	**PASSED**
Auditor	**Jarmo van de Seijp** https://tinyurl.com/Jvdseijp

RISK SUMMARY

Risk Level	Total	Found	Pending	Solved	Acknowledgde	Objected
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	3	3	0	2	1	0
Minor	1	1	0	0	1	0
Informative	37	37	0	0	37	0
Discussion	0	0	0	0	0	0

FINDINGS

Function Default Visibility

SWC-ID: SWC-100

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:
Functions that do not have a function visibility type specified are public by default. This can lead to a vulnerability if a developer forgot to set the visibility and a malicious user is able to make unauthorized or unintended state changes.

Category	Risk Level	Number of Findings	Status
SWC-100	Informative	**8**	Solved

Unused Code

Relationship:
CWE-1164: Irrelevant Code

Description:

Unused variables are allowed in Solidity and they do not pose a direct security issue. It is best practice though to avoid them as they can:

- cause an increase in computations (and unnecessary gas consumption)

- indicate bugs or malformed data structures and they are generally a sign of poor code quality

- cause code noise and decrease readability of the code

Category	Risk Level	Number of Findings	Status
Dead Code	informational	**17**	Solved

Multiple Pragma Directives used

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:

In Truffle or Hardhat projects, importing files with their original pragma directive is common practice. However, since this smart contract is a single-file contract, it is recommended to use 1 pragma directive at the top of the file.

Category	Risk Level	Number of Findings	Status
Pragma Directives	Informative	**12**	Solved

Missing Events

Description:

The critical variables currentStage and soldAmount (though soldAmount to a lesser extend) play an important role in the smart contract, since they are key players in its initial and subsequent ecosystem. The change of these variables is not emitted as an event. This may cause 3rd party applications as well as users to miss the change in price, causing unwanted outcome for users or aggregators

Category	Risk Level	Number of Findings	Status
Missing-Events	Medium	**2**	Solved

Push-Over-Pull

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:

The transfer of the contract's ownership through the function **transferOwnership()** only has 1 check, which is to ensure that the new owner is not the 0 address. It does not, however, check whether or not the ownership can be accepted by the recipient **newOwner**. In the case of a transfer of ownership to an incorrect address, or a smart contract that is not able to use the privileged functions, ownership of the contract is lost permanently with no way of getting it back. It is therefore advisable to use a pull method as opposed to push, in which case the **newOwner** would have to pro-actively accept ownership upon receiving it.

Category	Risk Level	Number of Findings	Status
Push over Pull	Minor	**1**	Acknowledged

Risk of Centralization

Description:

There is only 1 privileged role in the smart contract, owner, who controls every privileged function. If the account or owner is compromised or if accesss is lost, the project could suffer losses.

Category	Risk Level	Number of Findings	Status
Centralizaiton Risk	medium	**1**	Acknowledged

Note from the Author:

"
The client has major parts of the contract re-written to accommodate a more sustainable and secure version of the contract, written and audited by a certified party
"

AUDIT RESULT

Basic Coding Bugs

1. Constructor Mismatch

o Description: Whether the contract name and its constructor are not identical to each other.

o Result: PASSED

o Severity: Critical

Ownership Takeover

o Description: Whether the set owner function is not protected.

o Result: PASSED

o Severity: Critical

Redundant Fallback Function

o Description: Whether the contract has a redundant fallback function.

o Result: PASSED

o Severity: Critical

Overflows & Underflows

Description: Whether the contract has general overflow or underflow Vulnerabilities

o Result: PASSED

o Severity: Critical

Reentrancy

o Description: Reentrancy is an issue when code can call back into your contract and change state, such as withdrawing ETHs.

o Result: PASSED

o Severity: Critical

MONEY-Giving Bug

o Description: Whether the contract returns funds to an arbitrary address.

o Result: PASSED

o Severity: High

Blackhole

- *Description: Whether the contract locks ETH indefinitely: merely in without out.*
- *Result: PASSED*
- *Severity: High*

Unauthorized Self-Destruct

- *Description: Whether the contract can be killed by any arbitrary address.*
- *Result: PASSED*
- *Severity: Medium*

Revert DoS

- *Description: Whether the contractis vulnerable to DoSattack because of unexpected revert.*
- *Result: PASSED*
- *Severity: Medium*

Unchecked External Call

- *Description: Whether the contract has any external call without checking the return value.*
- *Result: PASSED*
- *Severity: Medium*

Gasless Send

- *Description: Whether the contractis vulnerable to gasless send.*
- *Result: PASSED*
- *Severity: Medium*

Send Instead of Transfer

- *Description: Whether the contract uses send instead of transfer.*
- *Result: PASSED*
- *Severity: Medium*

Costly Loop

o Description: Whether the contract has any costly loop which may lead to Out-Of-Gas exception.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Untrusted Libraries

o Description: Whether the contract use any suspicious libraries.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Predictable Variables

o Description: Whether the contract contains any randomness variable, but its value can be predicated.

o Result: PASSED

o Severity: Medium

Transaction Ordering Dependence

o Description: Whether the final state of the contract depends on the order of the transactions.

o Result: PASSED

o Severity: Medium

. Deprecated Uses

o Description: Whether the contract use the deprecated tx.origin to perform the authorization.

o Result: PASSED

o Severity: Medium