

# Assure DeFi™

The Verification **Gold Standard**™



## Security Assessment **Embark Token**

October 20, 2023

Audit Status: Fail

Audit Edition: Advance



ASSURE DEFI™  
THE VERIFICATION **GOLD STANDARD**

# Risk Analysis

## Classifications of Manual Risk Results

Classification	Description
<span style="color: red;">●</span> Critical	Danger or Potential Problems.
<span style="color: orange;">●</span> High	Be Careful or Fail test.
<span style="color: yellow;">●</span> Low	Pass, Not-Detected or Safe Item.
<span style="color: lightblue;">●</span> Informational	Function Detected

## Manual Code Review Risk Results

Contract Privilege	Description
<span style="color: green;">●</span> Buy Tax	5
<span style="color: green;">●</span> Sale Tax	5
<span style="color: green;">●</span> Cannot Sale	Pass
<span style="color: green;">●</span> Cannot Sale	Pass
<span style="color: green;">●</span> Max Tax	5
<span style="color: green;">●</span> Modify Tax	Not Detected
<span style="color: green;">●</span> Fee Check	Pass
<span style="color: green;">●</span> Is Honeypot?	Not detected
<span style="color: green;">●</span> Trading Cooldown	Not Detected
<span style="color: red;">●</span> Can Pause Trade?	Fail
<span style="color: red;">●</span> Pause Transfer?	Detected, Owner needs to enable trade.
<span style="color: green;">●</span> Max Tx?	Pass
<span style="color: green;">●</span> Is Anti Whale?	Not Detected
<span style="color: green;">●</span> Is Anti Bot?	Not Detected

Contract Privilege	Description
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not-Detected
Can Mint?	Fail
Is Proxy?	Not Detected
Can Take Ownership?	Not detected
Hidden Owner?	Not detected
Owner	0x95f9f350590d58231d0b8b8d91cb646e83f61cb8
Self Destruct?	Not Detected
External Call?	Not detected
Other?	Not detected
Holders	0
Auditor Confidence	Medium Risk

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

# Project Overview

## Token Summary

Parameter	Result
Address	0x440af7c75fC8Cf9ebD97f04ee2B10eD406F68039
Name	Embark
Token Tracker	Embark (XARK)
Decimals	18
Supply	0
Platform	Polygon
compiler	v0.8.19+commit.7dd6d404
Contract Name	EmbarkToken
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	<a href="https://polygonscan.com/address/0x440af7c75fC8Cf9ebD97f04ee2B10eD406F68039#code">https://polygonscan.com/address/0x440af7c75fC8Cf9ebD97f04ee2B10eD406F68039#code</a>
Payment Tx	0x96d9c13d372d871054d010770e63780881ee434053904305b1cba6eeb9eca084

## **Main Contract Assessed**

### **Contract Name**

Name	Contract	Live
Embark	0x440af7c75fC8Cf9ebD97f04ee2B10eD406F68039	Yes

## **TestNet Contract Assessed**

### **Contract Name**

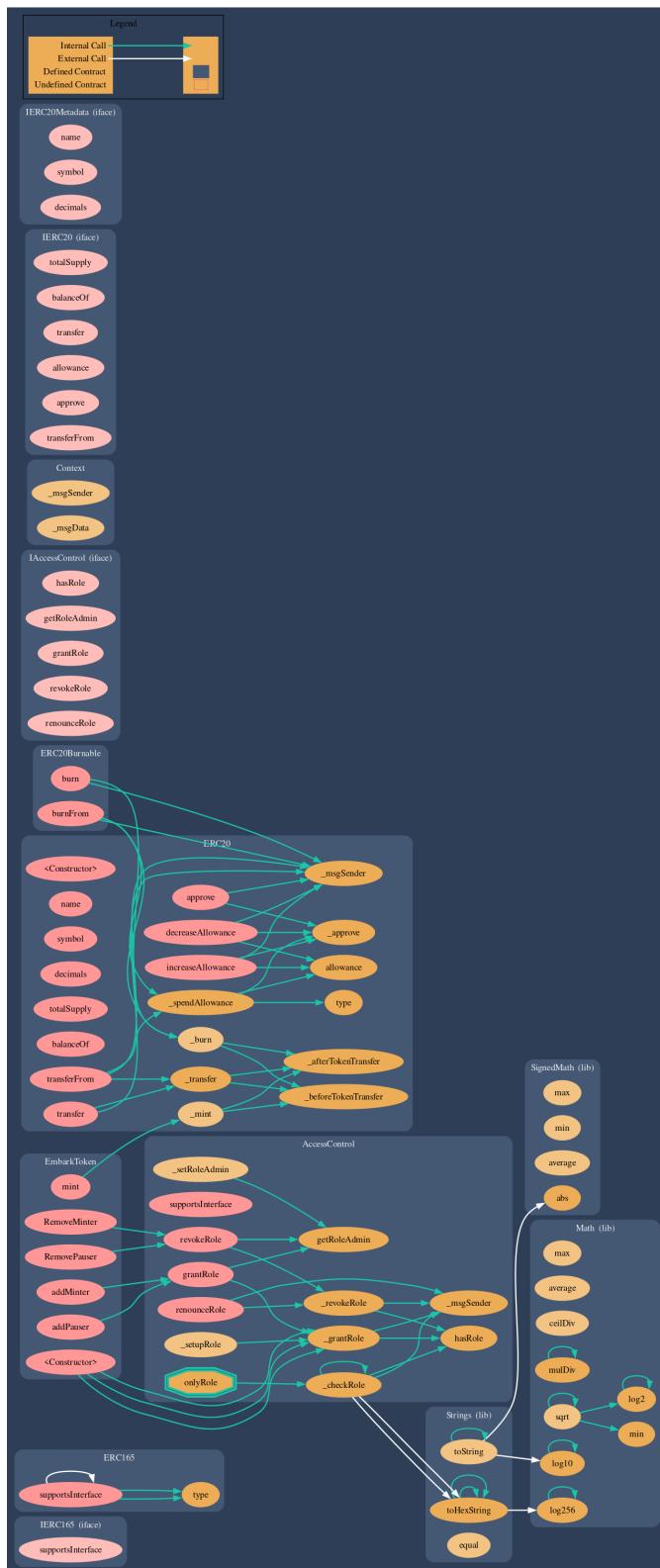
Name	Contract	Live
Embark	0x0eaBa785cDE03Ff7d1107fd887D29a0E5BeDd5b1	Yes

## **Solidity Code Provided**

SolidID	File Sha-1	FileName
Embark	222c7a1035ad660e8d98c2cbd200427270c1f61f	Embark.sol
Embark		
Embark		
Embark		

# Call Graph

The contract for Embark has the following call graph structure.



# Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	Embark.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	Embark.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	Embark.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	Embark.sol	L: 10 C: 0, L: 38 C: 0, L: 69 C: 0, L: 115 C: 0, L: 457 C: 4, L: 544 C: 4, L: 635 C: 4, L: 662 C: 4, L: 912 C: 4, L: 993 C: 4, L: 1023 C: 4, L: 1390 C: 4, L: 1430 C: 4
SWC-104	Pass	Unchecked Call Return Value.	Embark.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	Embark.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	Embark.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	Embark.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	Embark.sol	L: 0 C: 0

<b>ID</b>	<b>Severity</b>	<b>Name</b>	<b>File</b>	<b>location</b>
SWC-109	Pass	Uninitialized Storage Pointer.	Embark.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	Embark.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	Embark.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	Embark.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	Embark.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	Embark.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	Embark.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	Embark.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	Embark.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	Embark.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	Embark.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randomness.	Embark.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	Embark.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	Embark.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	Embark.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	Embark.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	Embark.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	Embark.sol	L: 0 C: 0

<b>ID</b>	<b>Severity</b>	<b>Name</b>	<b>File</b>	<b>location</b>
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	Embark.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	Embark.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	Embark.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	Embark.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	Embark.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	Embark.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	Embark.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	Embark.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	Embark.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	Embark.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

# **Smart Contract Vulnerability Details**

## **SWC-103 - Floating Pragma.**

### **CWE-664: Improper Control of a Resource Through its Lifetime.**

#### **References:**

#### **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.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

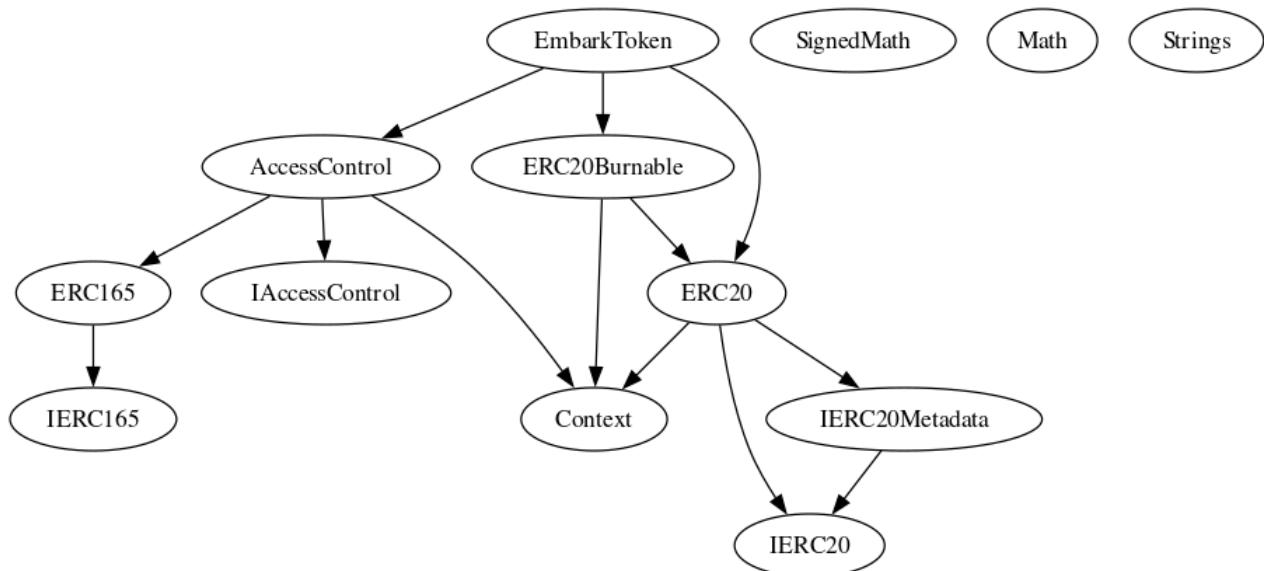
#### **References:**

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

# Inheritance

The contract for Embark has the following inheritance structure.

The Project has a Total Supply of 0



# Smart Contract Advance Checks

ID	Severity	Name	Result	Status
XARK-01	Low	Potential Sandwich Attacks.	Pass	Not Detected
XARK-02	Low	Function Visibility Optimization	Pass	Not Detected
XARK-03	Critical	Lack of Input Validation.	Fail	Detected
XARK-04	High	Centralized Risk In addLiquidity.	Pass	Not Detected
XARK-05	Critical	Missing Event Emission.	Fail	Detected
XARK-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not Detected
XARK-07	Low	State Variables could be Declared Constant.	Pass	Not Detected
XARK-08	Low	Dead Code Elimination.	Pass	Not Detected
XARK-09	High	Third Party Dependencies.	Pass	Not Detected
XARK-10	High	Initial Token Distribution.	Pass	Not Detected
XARK-11	Low	Multisend is present in code.	Pass	Not Detected
XARK-12	High	Centralization Risks In The X Role	Pass	Not Detected
XARK-13	Informational	Extra Gas Cost For User..	Pass	Not Detected
XARK-14	Critical	Unnecessary Use Of SafeMath	Pass	Not Detected
XARK-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
XARK-16	Medium	Taxes can be up to 100%	Pass	Not Detected
XARK-17	Informational	Conformance to numeric notation best practice.	Pass	Not Detected
XARK-18	Critical	Stop Transactions by using Enable Trade.	Fail	Detected, Owner needs to enable trade.

# XARK-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	 Critical	Embark.sol: 125,14	 Detected

## Description

The given input is missing the check for the non-zero address.

The given input is missing the check for the unSetPair is missing required function.

## Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
require(receiver != address(0), "Receiver is the zero address");
...
...
require(value X limitation, "Your not able to do this function");
...
```

We also recommend customer to review the following function that is missing a required validation. unSetPair is missing required function.

# XARK-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	 Critical	Embark.sol: 125, 14	 Detected

## Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes. The linked code does not create an event for the transfer.

## Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

## XARK-18 | Stop Transactions by using Enable Trade.

Category	Severity	Location	Status
Logical Issue	<span style="color: red;">●</span> Critical	Embark.sol: 469, 13	<span style="color: red;">[!]</span> Detected, Owner needs to enable trade.

### Description

Enable Trade is present on the following contract and when combined with Exclude from fees it can be considered a whitelist process, this will allow anyone to trade before others and can represent and issue for the holders.

### Remediation

We recommend the project owner to carefully review this function and avoid problems when performing both actions.

### Project Action

# Technical Findings Summary

## Classification of Risk

Severity	Description
<span style="color: red;">●</span> Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
<span style="color: orange;">●</span> High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
<span style="color: yellow;">●</span> Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
<span style="color: green;">◆</span> Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
<span style="color: blue;">ℹ</span> Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

## Findings

Severity	Found	Pending	Resolved
<span style="color: red;">●</span> Critical	1	0	0
<span style="color: orange;">●</span> High	0	0	0
<span style="color: yellow;">●</span> Medium	0	0	0
<span style="color: green;">◆</span> Low	2	0	0
<span style="color: blue;">ℹ</span> Informational	0	0	0
Total	3	0	0

# Social Media Checks

Social Media	URL	Result
Twitter	<a href="https://twitter.com/Embark_fi">https://twitter.com/Embark_fi</a>	Pass
Other	@embark.fi (instagram), <a href="https://discord.gg/8Y9X6HaPcX">https://discord.gg/8Y9X6HaPcX</a>	Pass
Website	<a href="https://www.embark.fi">https://www.embark.fi</a>	Pass
Telegram	<a href="https://t.me/embark_fi">https://t.me/embark_fi</a>	Pass

We recommend to have 3 or more social media sources including a completed working websites.

**Social Media Information Notes:**

**Auditor Notes:** undefined

**Project Owner Notes:**



# **Audit Result**

## **Final Audit Score**

Review	Score
Security Score	84
Auditor Score	80

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 85 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

## **Audit Fail**



# **Assessment Results**

## **Important Notes:**

- The following contract has a Mint function, this means the supply will increase.i
- The team is KYC, while KYC has a better rating. We don't recommend contracts that have a mint function, we recommend them to have a mint compliance and to be transparent with their users about the Mint Intent.

**Auditor Score =80**  
**Audit Fail**



# Appendix

## Finding Categories

### Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

### Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

### Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

### Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

### Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

### Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

### Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

### Coding Best Practices

ERC 20 Coding Standards are a set of rules that each developer should follow to ensure the code meets a set of criteria and is readable by all the developers.

# Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocacy for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or depreciation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.

