



CFG NINJA AUDITS

Security Assessment

Altom Swap

July 17, 2023

Audit Status: Pass

Audit Edition: Advanced

Project Overview

Token Summary

Parameter	Result
Address	0x
Name	AITom
Token Tracker	AITom (SwapToken)
Decimals	18
Supply	0
Platform	Ethereum
compiler	v0.8.19+commit.7dd6d404
Contract Name	SwapToken
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://etherscan.io/token/
Payment Tx	Corporate



Main Contract Assessed Contract Name

Name	Contract	Live
AITom	0x	Yes

TestNet Contract Assessed Contract Name

Name	Contract	Live
AITom	0x3c049dAd2193430a39C17bA71d75b798D93fc64A	Yes

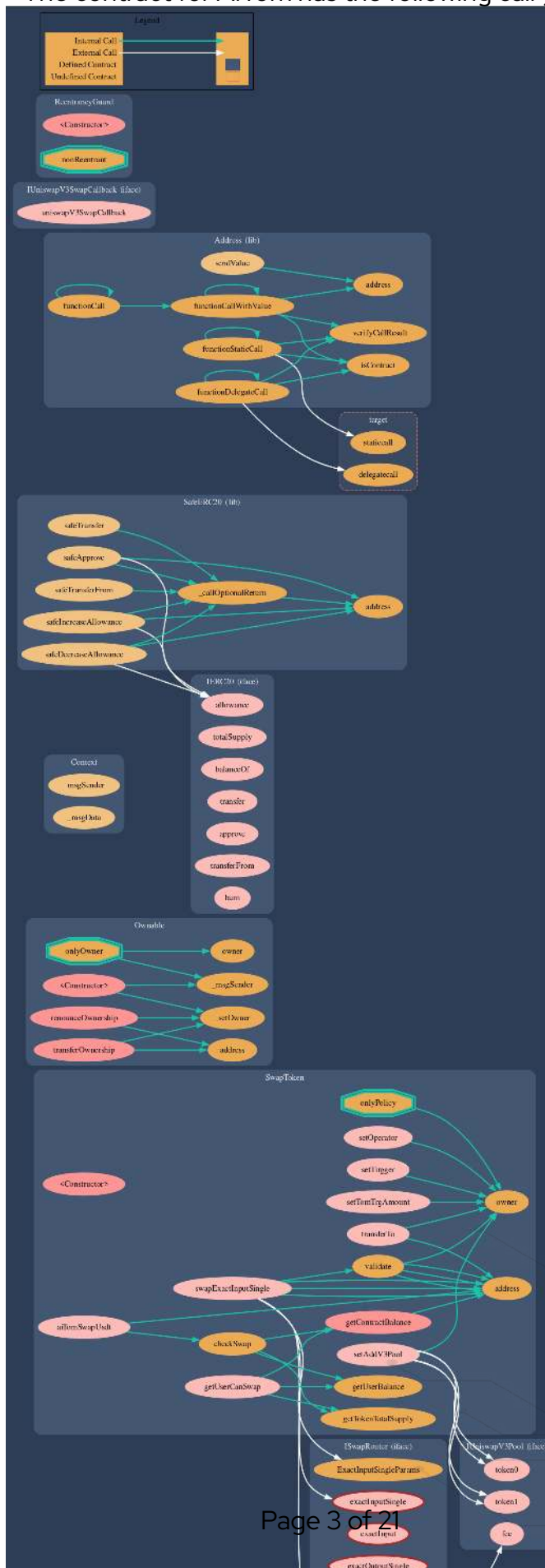
Solidity Code Provided

SolID	File Sha-1	FileName
AiTom	30fb0b9ae53f724bea6e2e36bab8e93bacc20a0a	SwapToken.sol
AiTom	68a6306f8c5cb8b28c2c1c9c8da1219998e86d3d	Ownable.sol
AiTom	41f987558cd34ad75ab7d03b61ae89bffd243ef	Context.sol
AiTom	9614db6b9e6b88f2df65ef00236137548d7256ed	SafeERC20.sol



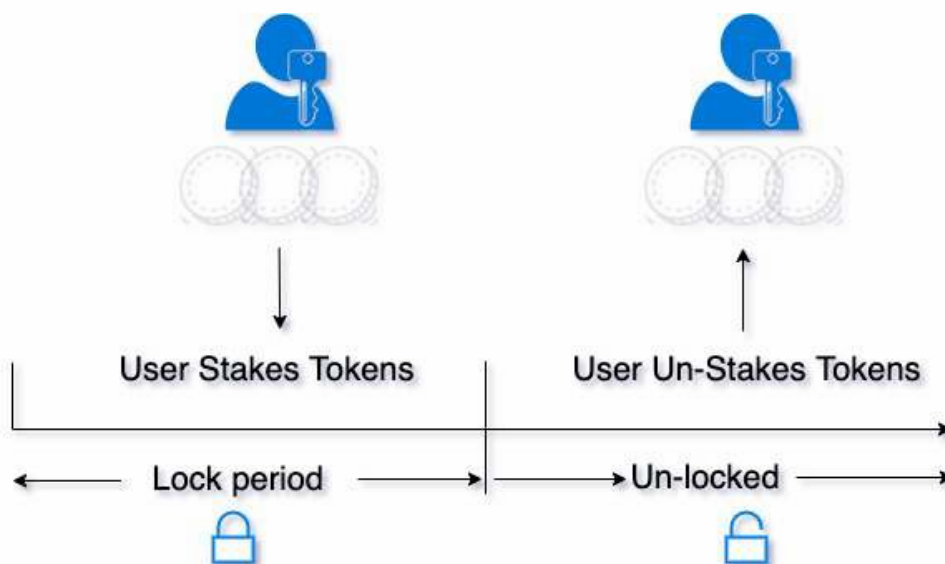
Call Graph

The contract for AITom has the following call graph structure.



What is a Staking Contract

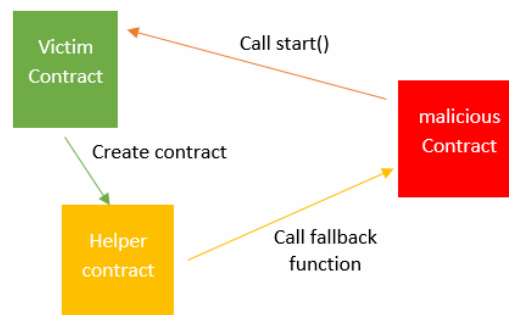
A smart contract which allows users to stake and un-stake a specified ERC20 token. Staked tokens are locked for a specific length of time (set by the contract owner at the outset). Once the time period has elapsed, the user can remove their tokens again.



The Project Owners of AlTom have implemented Reentrancy Guard Library

The Team has done a great job to avoid potential reentrancy issues in the contract.

You can read more about the reentrancy library used.
[ReentrancyGuard](#)



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	SwapToken.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	SwapToken.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	SwapToken.sol	L: 0 C: 0
SWC-103	Pass	A floating pragma is set.	SwapToken.sol	L: 0 C: 0
SWC-104	Pass	Unchecked Call Return Value.	SwapToken.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	SwapToken.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	SwapToken.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	SwapToken.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	SwapToken.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	SwapToken.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	SwapToken.sol	L: 0 C: 0



ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	SwapToken.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	SwapToken.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	SwapToken.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	SwapToken.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	SwapToken.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	SwapToken.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	SwapToken.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	SwapToken.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	SwapToken.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randommness.	SwapToken.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	SwapToken.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	SwapToken.sol	L: 0 C: 0
SWC-123	Low	Requirement Violation.	SwapToken.sol	L: 158 C: 15, L: 12 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	SwapToken.sol	L: 0 C: 0



ID	Severity	Name	File	location
SWC-125	Pass	Incorrect Inheritance Order.	SwapToken.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	SwapToken.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	SwapToken.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	SwapToken.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	SwapToken.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	SwapToken.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	SwapToken.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	SwapToken.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	SwapToken.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	SwapToken.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	SwapToken.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	SwapToken.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-123 - Requirement Violation

CWE-573: Improper Following of Specification by Caller

Description:

The Solidity `require()` construct is meant to validate external inputs of a function. In most cases, such external inputs are provided by callers, but they may also be returned by callees. In the former case, we refer to them as precondition violations. Violations of a requirement can indicate one of two possible issues:

- A bug exists in the contract that provided the external input.
- The condition used to express the requirement is too strong.

Remediation:

If the required logical condition is too strong, it should be weakened to allow all valid external inputs. Otherwise, the bug must be in the contract that provided the external input and one should consider fixing its code by making sure no invalid inputs are provided.

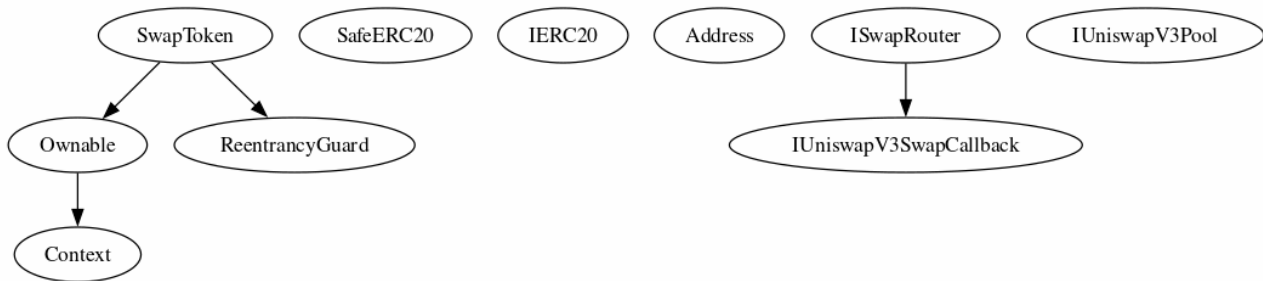
References:

The use of `revert()`, `assert()`, and `require()` in Solidity, and the new REVERT opcode in the EVM



Inheritance

The contract for AlTom has the following inheritance structure.



Privileged Functions (onlyOwner)

Please Note if the contract is Renounced none of this functions can be executed.

Function Name	Parameters	Visibility
renounceOwnership		Public
transferOwnership	address newOwner	Public
transferTo		External
setTomTrgAmount		External
setAddV3Pool		External
setTirgger		External
setOperator		External



Smart Contract Advance Checks



ID	Severity	Name	Result	Status
SwapToken-01	Low	Potential Sandwich Attacks.	Pass	Not Detected
SwapToken-02	Informational	Function Visibility Optimization	Pass	Not Detected
SwapToken-03	Low	Lack of Input Validation.	Fai	Detected
SwapToken-04	High	Centralized Risk In addLiquidity.	Pass	Not Detected
SwapToken-05	Low	Missing Event Emission.	Pass	Not Detected
SwapToken-06	Low	Conformance with Solidity Naming Conventions.	Pass	Not Detected
SwapToken-07	Low	State Variables could be Declared Constant.	Pass	Not Detected
SwapToken-08	Low	Dead Code Elimination.	Pass	Not Detected
SwapToken-09	High	Third Party Dependencies.	Pass	Not Detected
SwapToken-10	High	Initial Token Distribution.	Pass	Not Detected
SwapToken-11	High	claimStuckTokens can claim own tokens.	Pass	Not Detected
SwapToken-12	High	Centralization Risks In The X Role	Pass	Not Detected
SwapToken-13	Informational	Extra Gas Cost For User..	Pass	Not Detected



ID	Severity	Name	Result	Status
SwapToken-1 4	Medium	Unnecessary Use Of SafeMath	Pass	Not Detected
SwapToken-1 5	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not Detected
SwapToken-1 6	Medium	Taxes can be up to 100%	Pass	Not Detected
SwapToken-1 7	Logical Issue	Highly Permissive Role Access,`	Pass	Not Detected
SwapToken-1 8	Critical	Stop Transactions by using Enable Trade.	Pass	Not Detected



SwapToken-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	 Low	SwapToken.sol: L: 158 C: 15,L: 12 C: 0	 Detected

Description

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

The given input is missing the check for the allOnly Owners.

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. allOnly Owners.








Technical Findings Summary

Classification of Risk

Severity	Description
 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.
 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.
 Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
 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.
 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
 Critical	0	0	0
 High	0	0	0
 Medium	0	0	0
 Low	1	0	0
 Informational	0	0	0
Total	1	0	0



Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/MemeAitom	Pass
Other		Fail
Website	https://aitom.pro/	Pass
Telegram	https://t.me/AITomPro	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:



Assessment Results

Score Results

Review	Score
Overall Score	100/100
Auditor Score	95/100
Review by Section	Score
Manual Scan Score	43/33
SWC Scan Score	35 /37
Advance Check Score	35 /30

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 80 Points, if a project does not attain 80% is an automatic failure. Read our notes and final assessment below.

Audit Passed



Assessment Results

Important Notes:

- This is a custom router to swap token to usdt.
- Please DYOR on the project.

Auditor Score =95

Audit Passed



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 requirements 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

CFGNINJA 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 deprecation 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 CFGNINJA is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will CFGNINJA 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 CFGNINJA 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.

