DeHacker

Code Security Assessment

Morpheus Network Swapper

June10th, 2023

Contents

Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.
Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

Informational

A vulnerability that has informational character but is not affecting any of the code.

Overview

Project Summary

Project Name	Morpheus Network Swapper
Platform	Ethereum
Website	https://morpheus.network/
Type	Others
Language	Solidity

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	1	0	0	0	0	1
Minor	3	0	0	3	0	0
Informational	4	0	0	1	0	3
Discussion	0	0	0	0	0	0

Audit scope

ID	File	SHA256 Checksum
MNW	MNW_tokenswap.sol	66ec364bffa681931eb6ffdb155771fd4fb6a9f4d0a76a41571946db15a702b7

Findings

ID	Category	Severity	Status
MNW-01	Mathematical Operations	Informational	Resolved
MNW-02	Logical Issue	Medium	Resolved
MNW-03	Logical Issue	Minor	Acknowledged
MNW-04	Gas Optimization	Informational	Resolved
MNW-05	Centralization / Privilege	Minor	Acknowledged
MNW-06	Logical Issue	Minor	Acknowledged
MNW-07	Mathematical	Informational	Resolved
MNW-08	Gas Optimization	Informational	Acknowledged

Informational

MNW-01 | Non-Standard SafeMath

Category	Severity	Location	Status
Mathematical Operations	Informational	MNW_tokenswap.sol: 37~47	Resolved

Description

The SafeMath implementation utilized by the contract is non-standard and conducts inefficient checks.

Recommendation

We advise the OpenZeppelin implementation to be utilized instead as the linked one is inefficient andquestionable.

Medium

MNW-02 | Incorrect SafeERC20 Implementation

Category	Severity	Location	Status
Logical Issue	Medium	Medium	Resolved

Description

The SafeERC20 implementation used by the contract is non-standard and actually does not properlyevaluate ERC20 transfers

Recommendation

We recommend the OpenZeppelin implementation to be utilized. Alternatively, we advise the Morpheusteam to opportunistically evaluate the returned boolean by the ERC20 transfers as in the currentimplementation valid USDT transfers will fail.

Minor

MNW-03 | Permissive Authorization Model

Category	Severity	Location	Status
Logical Issue	Minor	MNW_tokenswap.sol: 72~78	Acknowledged

Description

The current authorization model renders all of its members equal in addition and removal rights to the group.

Recommendation

We advise the functions that add and remove members from the group to be privileged by another role asin the current system a single caller can overtake the system and set all others to false .

Informational

MNW-04 | Mutability Specifiers Missing

Category	Severity	Location	Status
Gas Optimization	Informational	MNW_tokenswap.sol: 120~121, 114~115	Resolved

Description

The linked variables are assigned only once, either during their contract-level declaration or during theconstructor 's execution.

Recommendation

For the former, we advise that the constant keyword is introduced in the variable declaration to greatlyoptimize the gas cost involved in utilizing the variable. For the latter, we advise that the immutablemutability specifier is set at the variable's contract-level declaration to greatly optimize the gas cost ofutilizing the variables. Please note that the immutable keyword only works in Solidity versions v0.6.5 andup.

Minor

MNW-05 | Arbitrary Fund Access

Category	Severity	Location	Status
Centralization / Privilege	Minor	MNW_tokenswap.sol: 163~166, 159~161	Acknowledged

Description

The controller role has arbitrary access to all ERC20 tokens held by the contract.

Recommendation

We advise this trait to be evaluated as it allows any tokens held by the contract to be siphoned out at will.

Minor

MNW-06 | Ineffectual Protection

Category	Severity	Location	Status
Logical Issue	Minor	MNW_tokenswap .sol: 154~157	Acknowledged

Description

The blacklist methodology utilized by the contract can be easily circumvented by generating a newcontract address and funelling funds through that one to the swap contract.

Recommendation

We advise the system to instead use a whitelist system that cannot be circumvented by a person with illicit funds.

Informational

MNW-07 | Incorrect Value Emitted

Category	Severity	Location	Status
Mathematical Operations	Informational	MNW_tokenswap.sol: 151	Resolved

Description

The amount of swapped tokens emitted by the function are incorrect.

Recommendation

We advise either the _amount to be used as is or multiplied by the offset rather than divided by it.

Informational

MNW-08 | Inexplicable Implementation

Category	Severity	Location	Status
Gas Optimization	Informational	MNW_tokenswap.sol: 134~136	Acknowledged

Description

The linked function serves no purpose in the contract.

Recommendation

We advise it to be safely omitted.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

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.

Coding Style

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

Volatile Code

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

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.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux “sha256sum" command against the target file.

About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

BLOCKCHAIINS

Ethereum
Cosmos
Eos
Substrate

TECH STACK

Python
Solidity
Rust
c++

CONTACTS

https://dehacker.io
https://twitter.com/dehackerio
https://github.com/dehacker/audits_public
https://t.me/dehackerio
https://blog.dehacker.io/

June 2023