Smart Contract Audit for Battlehard

Overlord Security

May 11, 2024

Contents

1 Project Overview **3**

2 Project Introduction **3**

3 Findings and Recommendations **3**

3.1 Summary 3

3.2 Critical Vulnerabilities 5

3.3 Medium Vulnerabilities 7

3.4 Low Vulnerabilities 8

3.5 Informational Vulnerabilities 10

4 Conclusion **15**

1 Project Overview

Created by: Battlehard

Based on: Neo Blockchain

Date Conducted: May, 2024

Battlehard hardened contract

Contracts: **hardened**
Github: `https://github.com/battlehard/hardened`
Commit: **675320f70a72**
Programming Language: **C#**
OS Env: **Neo 3.6.0**

2 Project Introduction

The idea for Battle Hardened was born out of the despair and loss that happens when a project creator pulls the rug, absconds as a leader, and leaves a community of token holders behind. The protocol seeks to remedy this despair by giving the NFT project's ownership back to the NFT holders while simultaneously giving agency to the project creators.

3 Findings and Recommendations

3.1 Summary

The following findings and recommendations after analyzing the **Battlehard hardened contract** implementation. Any additional recommendations beyond what any scanning tools supply are included as necessary.

Severity	Number of findings
Critical	2
Medium	1
Low	2
Informational	5

Issue Id	Severity	Title	Category	Fixed
MS-01	Critical	Insufficient permission verification	Coding Practices	Fixed
MS-02	Critical	Re-entrancy risk	Coding Practices	Fixed
MS-03	Medium	Potential key conflicts	Coding Practices	Fixed
MS-04	Low	non-compliance with the NEP-11	Coding Practices	Fixed
MS-05	Low	Lack of validity checks	Coding Practices	Fixed
MS-06	Informational	Typo	Optimization	Fixed
MS-07	Informational	Redundant code	Optimization	Fixed
MS-08	Informational	Overly Strict Checks	Business Logic	Fixed
MS-09	Informational	Potential key conflict	Business Logic	Fixed
MS-10	Informational	Inaccurate GAS calculation	Optimization	Fixed

3.2 Critical Vulnerabilities

MS-01: Insufficient permission verification

Insufficient permission verification

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db228
11d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Admin.cs#L18-
L31

Description

In a hardened project, the owner's permission is well checked by both Runtime.CheckWitness and the transaction's sender. While in contrast, the admin's permission is only checked by the tx.Sender. Noticed that on NEO blockchain, the tx.Sender's concept is somehow like the tx.origin on EVM-chains. We should not verify one's permission willing by only checking this sender cause any malicious dAPP could redirect a tx's execution path to anywhere.

Solution

It's recommended to replace the second return true to return Runtime.CheckwWtness(tx.Sender).

Status

The issue has been confirmed by the team and fixed in commit `e683d70`

MS-02: Re-entrancy risk

Re-entrancy risk

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db228 11d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Admin.cs#L18-L31

Description

In hardened project, the InfusionMint and InfusionUpdate as well as CancelInfusion didn't follow this pattern. Among all these functions, conditions are checked first, then the Safe11Transfer and Safe17Transfer and sometimes Mint are called, while the contract's storage is changed at the last step. All those token transfer could call unknown external code and may re-enter the hardened's contract. The attacker could drain the contract's NEP-17 token by re-enter the CancelInfusion.

Solution

It's recommended to change the PendingStorage's state before all those external calls. The functions could perform checks first as before, and then change the PendingStorage, and at the last step do all other interactions such as transfer NEP-11 and NEP-17 to other addresses.

Status

The issue has been confirmed by the team and fixed in commit `161f7f2`

3.3 Medium Vulnerabilities

MS-3: Potential key conflicts colback

Potential key conflicts

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db22811d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Storage.cs#L13-L19

Description

In hardened, we still use the prefix 0x01 and 0x02 for other purposes. If the keys concatenated after the prefix were not well checked, the real storage key could conflict with the internal Nep11Token's keys.

Solution

It's recommended to change the first prefix starting from 0x05 or 0x10.

Status

The issue has been confirmed by team and fixed in commit `e2e763b`.

3.4 Low Vulnerabilities

MS-04: Non-compliance with the NEP-11 proposal

Non-compliance with the NEP-11 proposal

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db22 811d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Admin.cs#L39-L43

Description

In hardened project, there exist some functions lack of validity checks.

Solution

It's recommended to perform ValidateScriptHash for contractHash. It's recommended to perform valid range checks for the non-null parameters such as positive-test for gasMintCost. It's recommended to perform checks to payTokenHash and payTokenAmount in InfusionUpdate like what the InfusionMint done.

Status

The issue has been confirmed by the team and fixed in commit `c5a0bd2`

MS-05:Non-compliance with the NEP-11 proposal

Non-compliance with the NEP-11 proposal

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db2281 1d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Helpers.cs#L20

Description

In hardened project, ValidateExternalNftOwnership function use the properties instead of ownerOf for getting a NFT's owner which didn't conform to the proposal.

Solution

It's recommended to call the ownerOf method instead as suggested by the proposal here.

Status

The issue has been confirmed by team and fixed in commit `9670bb8`

3.5 Informational Vulnerabilities

MS-06: Typo

Typo

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db22811d6d96
60cfc9f714f54f32bb6b5/src/Hardened/Hardened.cs#L159

Description

There exists typo in the code such as refun.

Solution

Replacing typo text.

Status

The issue has been confirmed by team and fixed.

MS-07: Redundant code

Redundant code

Source Code link

https://github.com/battlehard/hardened/issues/10

Description

There are a number of redundant codes that can be improved.

Solution

Most of those codes can be merged or deleted.

Status

The issue has been confirmed by the team and fixed in commit `a48d3d7`

MS-08: Overly Strict Checks

Overly Strict Checks

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db2281 1d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Helpers.cs#L37-L56

Description

In a hardened project, newly added NFTs to a bhNFT will be checked for duplicates among existing NFTs. While a unique NFT is defined by both it's ID and it's contract-hash..

Solution

Only checking the NFT's ID will be somehow strictly.

Status

The issue has been confirmed by team.

MS-09 Potential key conflict by malicious consensus node

Potential key conflict by malicious consensus node

Source Code link

https://github.com/battlehard/hardened/blob/f196c00db228
11d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.Storage.cs#L92-
L95

Description

On NEO platform, the random number could be manipulated by the consensus node now by specifying the block.nonce.

Solution

This risk will disappear automatically after NEO upgrades to a more secure random source.

Status

The issue has been confirmed by team.

MS-10 Potential key conflict by malicious consensus node

Potential key conflict by malicious consensus node

Source Code link

https://github.com/battlehard/hardened/blob/f196c00d b22811d6d9660cfc9f714f54f32bb6b5/src/Hardened/Hardened.cs#L130-L132

Description

If the admin call CancelInfusion for multiple users in one transaction, the GAS will be miscalculated and counted multiple times. If not, the admin will have to afford the NetworkFee and a few SystemFee before this function's execution.

Solution

Therefore, it's recommended to use the ((Transaction)Runtime.ScriptContainer).SystemFee + ((Transaction)Runtime.ScriptContainer).NetworkFee and make sure only perform CancelInfusion for one user at once.

Status

The issue has been confirmed by team and fixed in commit `1d06f9d`.

4 Conclusion

In this audit, we have analyzed the **Battlehard hardened** design and implementation. The current code base is well organized and those identified issues are promptly confirmed and fixed.

Meanwhile, we need to emphasize that smart contracts as a whole are still in an early, but active stage of development. To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, audit findings, or potential gaps in scope/coverage.

For more information regarding this audit report, please send email to contact@overlord.wtf