EPNS Protocol 1.5 Smart Contracts Review

By: ChainSafe Systems

November 2022

EPNS Protocol 1.5 Smart Contracts Review

Auditors: Tanya Bushenyova, Oleksii Matiiasevych, Anderson Lee

WARRANTY

Introduction

EPNS requested ChainSafe Systems to perform a review of the EPNS Protocol 1.5 smart contracts. The contracts can be identified by the following git commit hash:

`3619f703fceb76ef54821f02736a42f2b1c3e224`

There are 4 contracts in scope, specifically EPNSCoreV1_Temp, EPNSCoreV1_5, EPNSCommStorageV1_5, EPNSCommV1_5. After the initial review, EPNS team applied a number of updates which can be identified by the following git commit hash:

`e1c682fe2a5fc2658bd3f73c6984a18d71b083b1`

Additional verification was performed after that.

Disclaimer

The review makes no statements or warranties about the utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about the fitness of the contracts for any specific purpose, or their bug free status.

Executive Summary

All the initially identified, minor and above, severity issues were fixed and are not present in the final version of the contracts. No new issues were discovered in the final version.

There are no known compiler bugs for the specified compiler version (0.6.11), that might affect the contracts' logic.

There were 0 critical, 1 major, 1 minor, 23 informational/optimizational issues identified in the initial version of the contracts. All the issues found in EPNS contracts were not present in the final version of the contracts. They are described below for historical purposes.

Critical Bugs and Vulnerabilities

No critical issues were identified.

Line by Line Review. Fixed Issues

1. EPNSCommV1_5, line 3: Note, `hardhat console` import could be removed.

2. EPNSCommV1_5, line 22: Note, the `IERC20` import could be removed.

3. EPNSCommV1_5, line 24: Note, the `SafeERC20` import could be removed.

4. EPNSCommV1_5, line 30: Note, the `SafeERC20` and `IERC20` could be removed.

5. EPNSCommV1_5, line 119: Note, the `transferPushChannelAdminControl()` function could be made `external`.

6. EPNSCommV1_5, line 148: Note, the `isUserSubscribed()` function has not consistent code style. In other functions, returned `boolean` params are not named.

7. EPNSCommV1_5, line 242: Optimization, the `_subscribe()` function reads `user.subscribedCount` from storage multiple times. Consider storing it in memory.

8. EPNSCommV1_5, line 261: Note, the `subscribeBySig()` function could be made `external`.

9. EPNSCommV1_5, line 371: Optimization, some fields of the user struct are read from storage multiple times in the `_unsubscribe()` function. Consider storing it in memory.

10. EPNSCommV1_5, line 398: Note, the `unsubscribeBySig()` function could be made `external`.

11. EPNSCommV1_5, line 604: Note, the `sendNotification()` function could be made `external`.

12. EPNSCommV1_5, line 667: Note, the `sendNotifBySig()` function could increase `nonces[_signer]` and return false.

13. EPNSCoreV1_Temp, line 14: Note, the `IPUSH` import could be removed.

14. EPNSCoreV1_Temp, line 248: Note, the `transferPushChannelAdminControl()` function could be made `external`.

15. EPNSCoreV1_5, line 26: Note, `hardhat console` could be removed.

16. EPNSCoreV1_5, line 265: Note, the `transferPushChannelAdminControl()` function could be made `external`.

17. EPNSCoreV1_5, line 326: Optimization, the `updateCounter` local variable could be used instead of `channelUpdateCounter[_channel]` in the `updateChannelMeta()` function.

18. EPNSCoreV1_5, line 471: Optimization, the `_createChannel()` function reads `channelsCount` from storage twice.

19. EPNSCoreV1_5, line 521: Optimization, some fields of the `channelData` struct are read from storage multiple times in the `destroyTimeBoundChannel()` function. Consider storing it in memory.

20. EPNSCoreV1_5, line 559: ***Major***, the `destroyTimeBoundChannel()` function deletes `channels[msg.sender]` instead of `channels[_channelAddress]`.

21. EPNSCoreV1_5, line 636: Note, in the `deactivateChannel()` function, the `_newChannelWeight` variable could be set equal to `ADJUST_FOR_FLOAT`.

22. EPNSCoreV1_5, line 679: Note, the `reactivateChannel()` function could revert with underflow if `FEE_AMOUNT > ADD_CHANNEL_MIN_FEES`.

23. EPNSCoreV1_5, line 730: Note, in the `blockChannel()` function, the `_newChannelWeight` variable could be set equal to `ADJUST_FOR_FLOAT`.

24. EPNSCoreV1_5, line 758: Note, `onlyActivatedChannels` modifier could be used in the `transferChannelOwnership()` function instead of checking channel state for readability.

25. EPNSCoreV1_5, line 798: Minor, the `transferChannelOwnership()` function should unsubscribe from other subscriptions as well, otherwise transferring back will revert.

Tanya Bushenyova

Oleksii Matiiasevych

Anderson Lee