Security and Design audit
Decentraland Avatar Name Registry

03/06/2019

Agustín Aguilar

Introduction

The Decentraland Team requested the review of the project on the repository https://github.com/decentraland/avatars-contract, the commit referenced for this audit is 0x99459ab45f8751174d10d4e57d53735bba088488.

The audited contracts are:

- AvatarNameStorage.sol: Defines the storage layout for AvatarNameRegistry.sol.
- AvatarNameRegistry.sol: Registers the ownership and metadata of a given Username, handles the process of allocating them.

Issues

High severity

1 - Visual spoofing vector in the username

Any user could register a username that looks almost identical to the username of another user, using visual spoofing, for example. The letter g U+0067 (LATIN SMALL LETTER G) looks almost identical to the letter ɡ U+0261 (LATIN SMALL LETTER SCRIPT G), so the usernames:

a - gonzalo1234
b - ɡonzalo1234

Would look identical but they are both valid usernames.

Proposed solutions:
Replace the character blacklist with a whitelist, and ranges can be used to avoid boilerplate code.

Update c61084b: The Decentraland team fixed the issue by restricting the registering of new usernames and only allow pre-authorized addresses.

Medium severity

2 - Username validation is too permissive for intended purposes.

The only requirement to consider a given username as valid is not to contain the U+0020 character (SPACE), but alternative but similar in characters are valid, for example, U+c2a0 (NO-BREAK SPACE) or U+0008 (BACKSPACE).

Proposed solution:
Replace the character blacklist with a whitelist, ranges can be used to avoid boilerplate code.

Update c61084b: The Decentraland team fixed the issue by restricting the registering of new usernames and only allow pre-authorized addresses.

Low severity

3 - Usernames front running

The avatar register contains a commit - reveal scheme to avoid front running of usernames, but there is no cost or limit on how many commits can be created, an attacker could pre-register all popular or existing usernames, and front run them when they are about to be registered.

It exists a limitation on only create one commit per address, but this limit can be easily circumvented using smart contracts.

Proposed solutions:

a - Perform the payment during the commit call, and not during the reveal.

b - Make commits expirable, making non-viable to maintain an extensive database of “sleeping” commits.

Update 87d1068: The Decentraland team fixed this issue using the proposed solution B.

Update c61084b: The Decentraland team removed the commit and reveal scheme.

4 - Storage inflation using commits

Any user can make use of the *commitUsername* to register values in the storage of the contract, currently, this is not an issue, but it may become one if any of the storage-rent proposals (https://github.com/ethereum/EIPs/issues/1418) gets implemented.

Proposed solution:

a - Store the commit in a different contract, created only to store one commit.

b - Charge the cost of registering a username during the commit call, and not during the reveal.

Update c61084b: The Decentraland team removed the commit and reveal scheme.

5 - Non msg.sender token pull (transferFrom)

During the reveal process the MANA tokens are always pulled directly from the beneficiary balance. This is dangerous, because if one approved account gets compromised, it could use funds of all account that approved the AvatarNameRegistry.

Proposed solution:

Only pull tokens from the msg.sender account.

Notes

6 - Misleading name of the internal _regiserUsername method

The internal method _regiserUsername not only registers the username, but it also charges a fee and burns tokens. Consider giving it a more specific name or splitting the functionality into multiple methods.

7 - Hardcoded price value

The price value is hardcoded in the contract, consider making it a constructor parameter or a constant.

8 - Misleading username length revert message

The revert message on line 223 states that the username should have at maximum 32 characters, but is actually validating if the username has less or equal 32 bytes.

If a user is using characters of more than 1 byte, the limit in chars will be lower.

Final thoughts

The username contracts are well written, documented, and have unit testing with good coverage; no critical vulnerabilities or bugs were found on AvatarNameStorage.sol or AvatarNameRegistry.sol.

June 2019, Madrid, Spain - by Agustin Esteban Aguilar