Audit Report

Project : AuditorDAO/**practice-audit.subscriptions**

Auditor: **zzzuhaibmohd(https://github.com/zzzuhaibmohd)**

Severity	Title
High	1. Any user can become an authorized signer for a user
High	2. `checkAuthorizedSigner` is defined but never used
Low	1. Missing unit tests for `collect`

[H-1] Any user can become an authorized signer for a user

`addAuthorizedSigner` allows a user to add others users as trusted signers for the users' behalf.

There is missing access control check to verify if the `msg.sender == _user`. As a result, anyone can become a signer on behalf of other user. Similar issue can be seen in `removeAuthorizedSigner` where in anyone can remove an authorized signer on behalf of an user.

Proof of Concept - Run the following Hardhat Test to verify the issue

```
it('Exploit#1 -> anyone can call authorizedSigners', async function () {
      const user_1 = subscriber1.address;
      const user_2 = subscriber2.address;
      const signer = await ethers.getSigner(user_1);

      await subscriptions.connect(signer).addAuthorizedSigner(user_2, user_1);
      expect(await subscriptions.checkAuthorizedSigner(user_2, user_1)).to.eq(
        true
      );
    });
```

Fix: Add the following check in `addAuthorizedSigner` and `removeAuthorizedSigner`

```
require(_user == msg.sender, 'caller and _user are different, Not Authorized!');
```

[H-2] `checkAuthorizedSigner` is defined but never used

`subscribe`, `extendSubscription`, `fulfil` are functions which can be called by self or any user. Who defines the anyone should be controlled by user via `checkAuthorizedSigner` user can control who can really call function on behalf of them.

Proof of Concept

```
it('Exploit#2 -> will allow anyone to start and extend a subscription', async function () {
      const now = await latestBlockTimestamp();
      const start = now;
      const end = now.add(1000);
      const newEnd = now.add(2000);
      const rate = BigNumber.from(5);
      const user = subscriber2.address;
      const subscribeBlockNumber = await subscribe(
        stableToken,
        subscriptions,
        subscriber1,
        start,
        end,
        rate,
        user
      );
      await mineNBlocks(150);
      await extendSubscription(
        stableToken,
        subscriptions,
        subscriber1,
        user,
        newEnd,
        subscribeBlockNumber
      );
    });
```

Fix: Add a check for in the following function to disallow unknown users from invoking these functions.

```
require(checkAuthorizedSigner(_user, msg.sender), "Not Authorized Signer");
```

[L-1] Missing unit tests for collect()

The code coverage report for `collect` function which is protected by `onlyOwner` modifier is missing. Adding unit tests for all the functions in the smart contract helps devs and auditors better understand the code.

Fix: Add missing unit tests for `collect`

Some Question is Mind, not sure if vulnerability

1. If user X **subscribes** on behalf of user Y. When `unsubscribe` is called the unlocked funds transferred by user X are returned to user Y. Moreover user X cannot call unsubscribe on behalf of user Y. Not sure if this functionality or a a bug.
2. Not really sure what is happening inside the while loop of `collect`, unit test would have helped better understand the functionality.