HakoDApp

Steps for Installation

1) Create a new directory named hakodapp

2) Clone ncipfs repository into into the hakodapp directory(https://github.com/drbh/ncipfs)

3) clone hako repository into the hakodapp directory(https://github.com/drbh/hako)

4) Change the directory to hako and enter the following commands

i. pipenv install --skip-lock-pre(Used to Ignore the Pipfile.lock and install from the Pipfile)

ii. pipenv shell(creates a virtual environemnt for the application)

iii. open a new terminal tab and direct to hakodapp/hako/app .

iv. run npm install (Installs necessary modules listed as dependencies in package.json)

v. run npm serve (Starts the UI server)

vi. you may have to edit the local IP address in app/src/App.vue and in app.py to your computers local IP (if running in offline mode) otherwise change the IPs accordingly to the networks you want to connect to.

vii. Run python3 app.py in the first terminal where the virtual environemnt is created.

Tips

- NCIPFS and hako repository should be in the same directories
- User needs to edit the local IP address in app/src/App.vue and in app.py to their computers local IP.
- Password should be more than 16 characters
- The port used should be your ipfs node.
- NCIPFS library is a must for Hako to run.

Errors received

At first, i got "ConnectionRefusedError: Host localhost:11501 Refused Connection", so i modified nucypher_network value as "52.14.207.225:9151". (modified the ipfs_api_gateway as 127.0.0.1:8080 (my ipfs node) , and modified var host = "192.168.1.3" as var host = "127.0.0.1" in app/src/App.vue file.

Another error I faced was ssl.SSLError: [SSL: WRONG_VERSION_NUMBER] wrong version number (_ssl.c:1056) and the problem was that The app was trying to access the server through http in place of https. I went to code and than changed it.

Entering the wrong password will not throw any errors until you attempt to upload something

Refreshing the page and logging in with another User clears the file list. Files are only persisted if the same user is logged in as the previous user (for that browser). This is since all decrypted file data is stored in the browsers local memory. Supporting that user's decrypted files are never shared outside of the local machine.

Last error was that for an existing username the server was not able to complete the key gen, and fails . I got stuck on the "Please wait while the Futari is being created" page

Application Critique

Hako is a decentralized file sharing application powered by NuCypher and IPFS. It uses a the proxy re-encryption scheme provided by NuCypher in order to secure and share data. The good part about it is that the user is always in control of credentials unlike centralized file shraing services such as Google Drive and Dropbox. Decenteralization provided by IPFS also makes it resistant to single point of failure. Usage of NuCypher also saves encryption as well as computational run time making the DApp more favorable. The hako code can still be modified by embedding the functionalties of NCIPFS library in the hako code itself making it completely independent.It basically handles NUCID'S and futari's(Users) NCIPFS library combines CID which is the content identifier for the data, the policy key created by the user for encryption as well as filename and encryption key. Therefore, on overall basis we can say that Hako is a very useful DApp whihc has the power to replace centralized file sharing applications providing security and tranparency to its users.