KeePassXCIPFS Script

Student Name : Jayal Shah
Capstone Project

Good Morning !!

My name is Jayal Shah and I am going to demonstrate on KeePassXCIPFS, which is a decentralized password manager developed by combining KeePassXC (Password manager) , IPFS (Decentralized storage), Nucypher (Re-encryption technology) and Syncthing (Synchronization tool).

The Password manager basically serves the purpose of sharing entire password databases as well as selected passwords among multiple users or nodes without disclosure of private credentials.

(VERBAL CUE)

ACTION : Opens KeePaasXC application and Database Name & Description > Decryption time and database format settings > Password selection > Location to save the database

Then explaining about adding and editing entries and certain features provided by the password manager.

Close the newly created database and open an already existing database consisting thousands of passwords for demo purpose.

(VERBAL CUE)

In the application we combined IPFS and Nucypher. IPFS is used as a decentralized storage while NuCypher is used for proxy re-encryption. Re-encryption basically provides 2 advantages. Firstly, it lets the private keys to be with the user itself. Secondly, it also saves network resources.

ACTION : Opens terminal one and starts IPFS daemon.

ACTION: Opens another terminal tab and starts nucypher ursula node as well as fleet of ursula in another tab.

ACTION: Then opens 3 new terminal tabs each representing different users or nodes. All the 3 nodes are then connected to the nucypher ursula network.

For demonstration purpose functions with parameters are called for each user whether to upload or download a file. In the given scenario user 1 encrypts and uploads the file and user 2 and user 3 download and decrypt the file.

User 1

- Generates policy public key.
- Selects the file encrypts it and uploads it to IPFS.
- Generated hash is provided to the Users 2 and 3.

User 2 & User 3

- Generates public and private keys.
- The Public key is then supplied to the ursula to generate policy_info.

Individual Policy_info's re-encrypted by the ursula nodes is supplied back to User 2 and user 3, using which the users download and decrypt the files with the help of their priv keys.

(VERBAL CUE)

Synchronization

Starting syncthing and exchanging ID'S among the users to connect with each other. Placing the file to be shared in the synchronization folder. Any user who wants to make changes to the password database can upload the edited copy to the synchronization folder. The password file gets synchronized for connected peers. If other user wants to edit the copy they can copy the file from synchronization folder edit it and move the edited file to synchronization folder for changes in other peers to take place.

(VERBAL CUE)

When talking about synchronization Cap theorem comes into play which states that decentralized database, system can only comply with 2 of the 3 : Consistency, availability and partition tolerance. Since partition tolerance is necessary as failure of a single node can not let the whole network go down. Therefore we are left with availability and consistency. In our case we comply with consistency rather than availability as all nodes in the network cannot be online all the time. In such cases the users might face certain issues as what if the a specific node updates the file and then again another node re-edits the file. There might be a node in the network that would not be available during the first edit as now directly needs to deal with second edit. Th only solution to cope with the problem of availability is to add separate files to with updates to the synchronization folder. In this way the offline node can even get the earlier update.

Keeshare

The application also provides feature to share selected passswords. For this the user needs to create a group for the passwords that they wish to share. Next we need to select edit group and go to keeshare feature and select export and enter password for the same. It tan creates a compressed folder with selected password file as well as users name, key , certificate as well as fingerprint to verify the user. After which the receiver can import the group consisting of selected passwords.

Questions ??

Thank You!!