DIGITAL IDENTITY VERIFICATION

Blockchain and Cryptocurrency Technologies

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DIGITAL ASSIGNMENT 2

SUNEHA GHOSH (18BCB0075)
SIDDHARTHA MONDAL (18BCB0145)

SUBMITTED TO:
PROF. BOOMINATHAN P.

SIDDHARTHA MONDAL

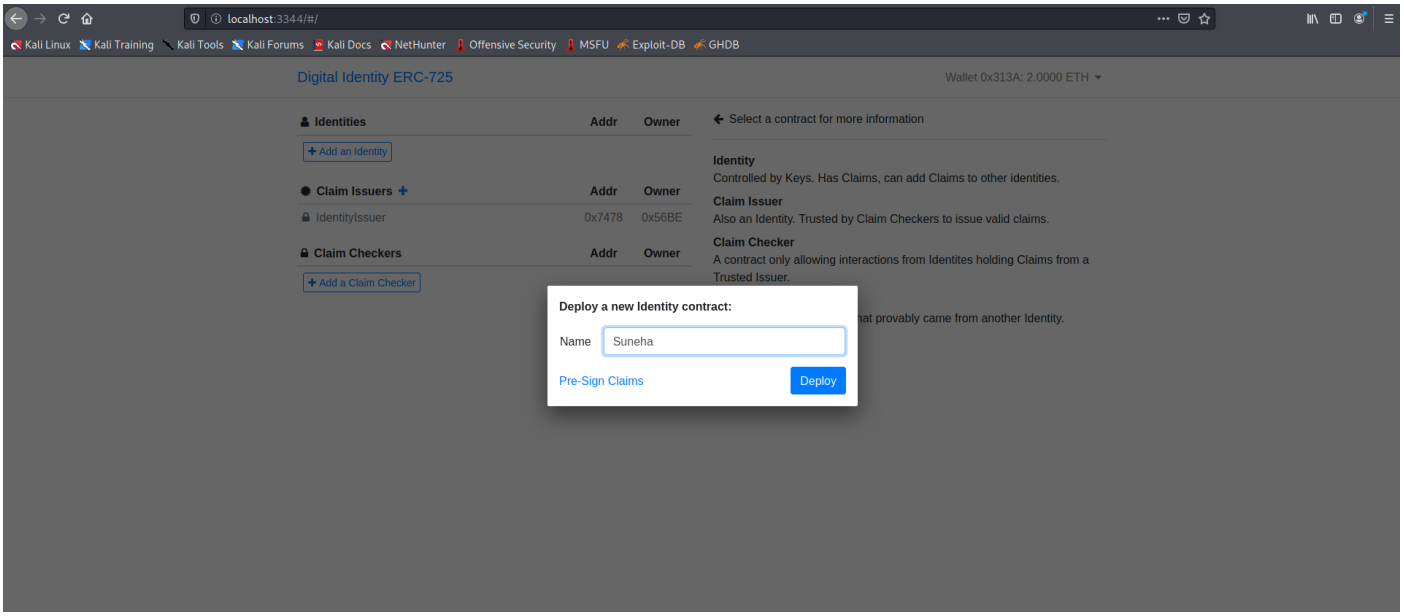
IMPLEMENTATION

1. We have implemented the complete project as root. For Starting the server we will first give the command

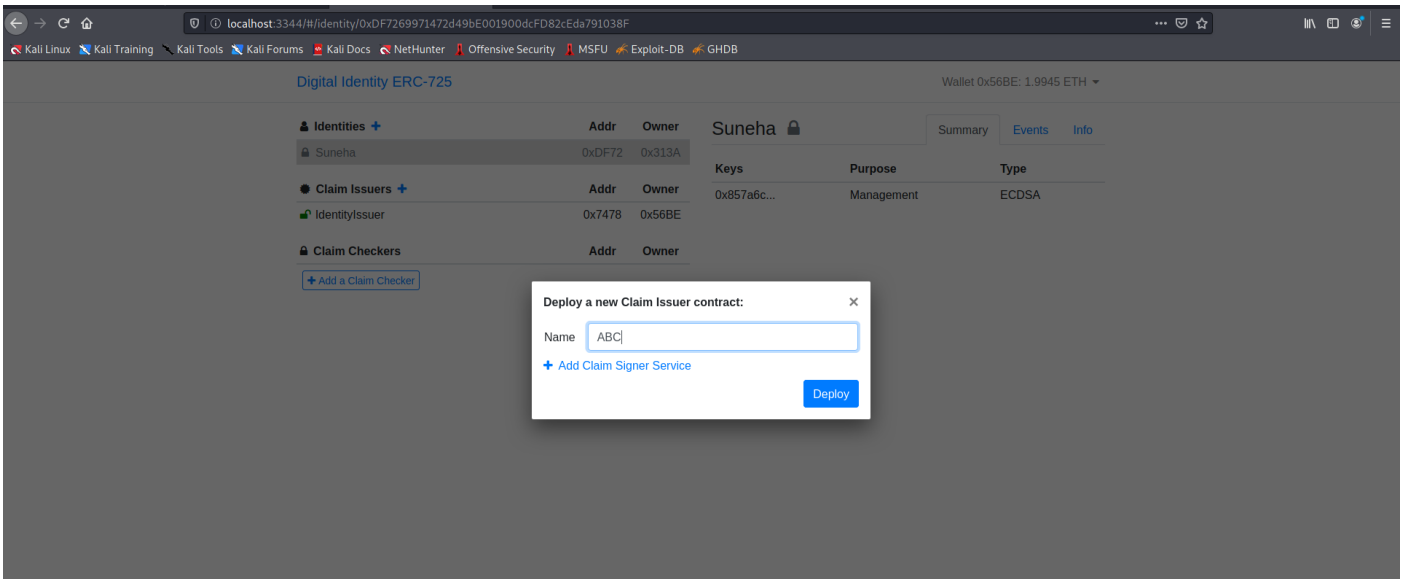
- nvm use 9.11.1 && yarn clean && yarn start

```
root@Suneha: /home/suneha/blockchain-identity
File  Actions  Edit  View  Help
^C

┌──(root💀 Suneha)-[/home/suneha/blockchain-identity]
└─# nvm use v9.11.1 && yarn clean && yarn start                                    130 ✗
Now using node v9.11.1 (npm v5.6.0)
yarn run v1.22.5
$ rm -rf data/db
Done in 0.23s.
yarn run v1.22.5
$ node -r @babel/register index
Browserslist: caniuse-lite is outdated. Please run next command `yarn upgrade caniuse-lite browsersl
ist`
Ganache listening. Starting webpack ...

Listening on host localhost, port 3344

ℹ ｢wds｣: Project is running at http://0.0.0.0:8080/
ℹ ｢wds｣: webpack output is served from /
Browserslist: caniuse-lite is outdated. Please run next command `yarn upgrade caniuse-lite browsersl
ist`
Opening Browser at http://localhost:3344
ℹ ｢wdm｣: wait until bundle finished: /vendor.js
ℹ ｢wdm｣: wait until bundle finished: /app.js
ℹ ｢wdm｣: Hash: ded83015
Version: webpack 4.20.2
Time: 17032ms
Built at: 2021-05-26 22:27:06
        Asset      Size  Chunks             Chunk Names
       app.js    478 KiB     app  [emitted]  app
    vendor.js   2.01 MiB  vendor  [emitted]  vendor
   app.js.map    317 KiB     app  [emitted]  app
vendor.js.map   2.22 MiB  vendor  [emitted]  vendor
Entrypoint app = vendor.js vendor.js.map app.js app.js.map
[./node_modules/loglevel/lib/loglevel.js] 7.68 KiB {vendor} [built]
[./node_modules/react-router-dom/es/index.js] 1010 bytes {vendor} [built]
[./node_modules/react-styl/index.js] 629 bytes {vendor} [built]
[./node_modules/react/index.js] 190 bytes {vendor} [built]
```

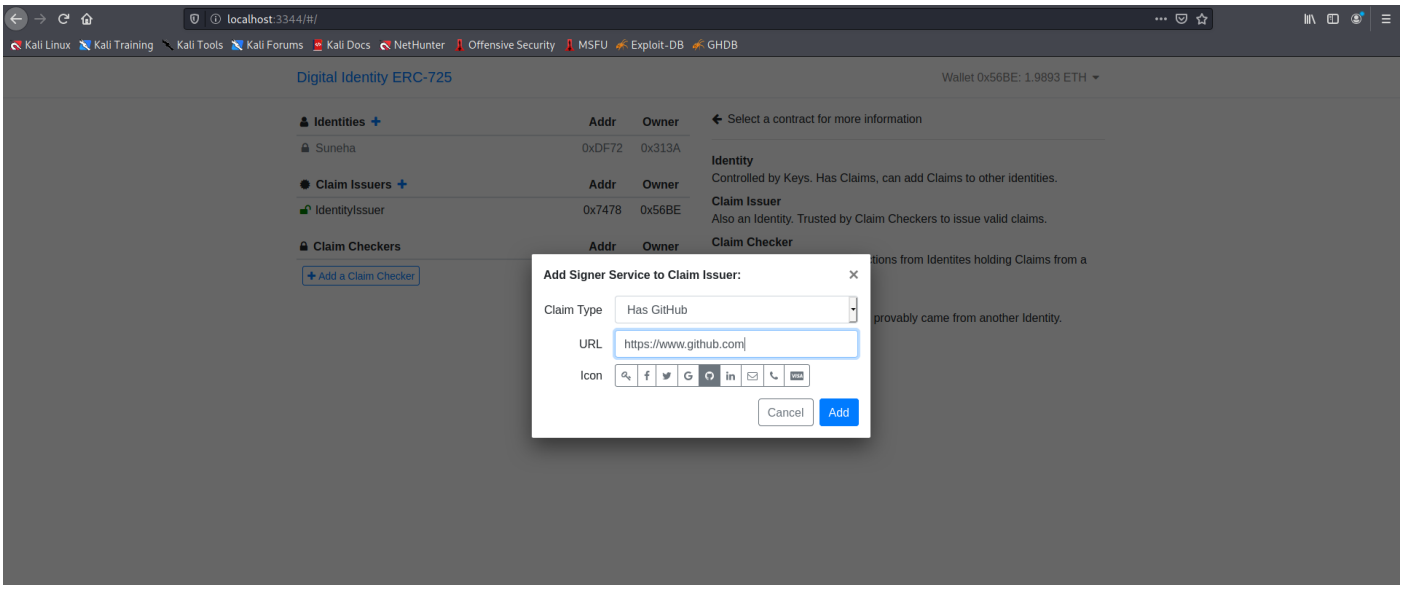
2. We can see the server host is http://localhost:3344
3. In the home page we can see Identities, Claim Issuers and Claim checkers.
4. We will first add a identity i.e. the person who wants to buy the property. After giving a name we will deploy it. Meanwhile the buyer has a wallet ID 0x313A



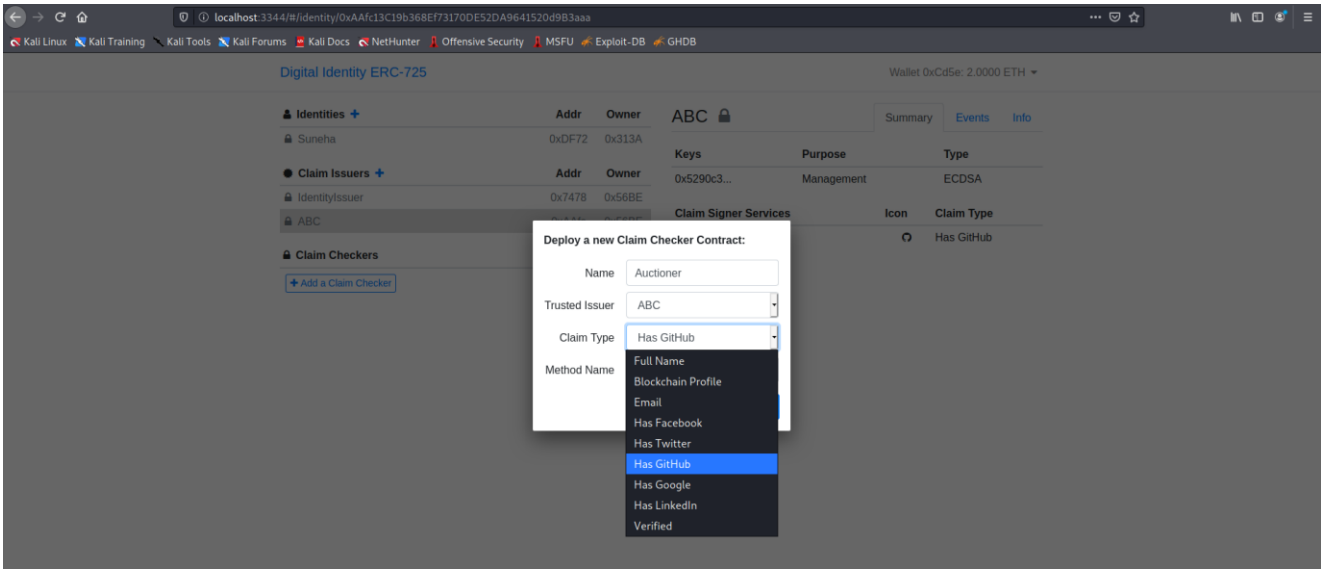
5. Then we will change the wallet to 0x56BE and add a claim issuer. The claim issuer who can verify the buyer's claim by issuing him claims.

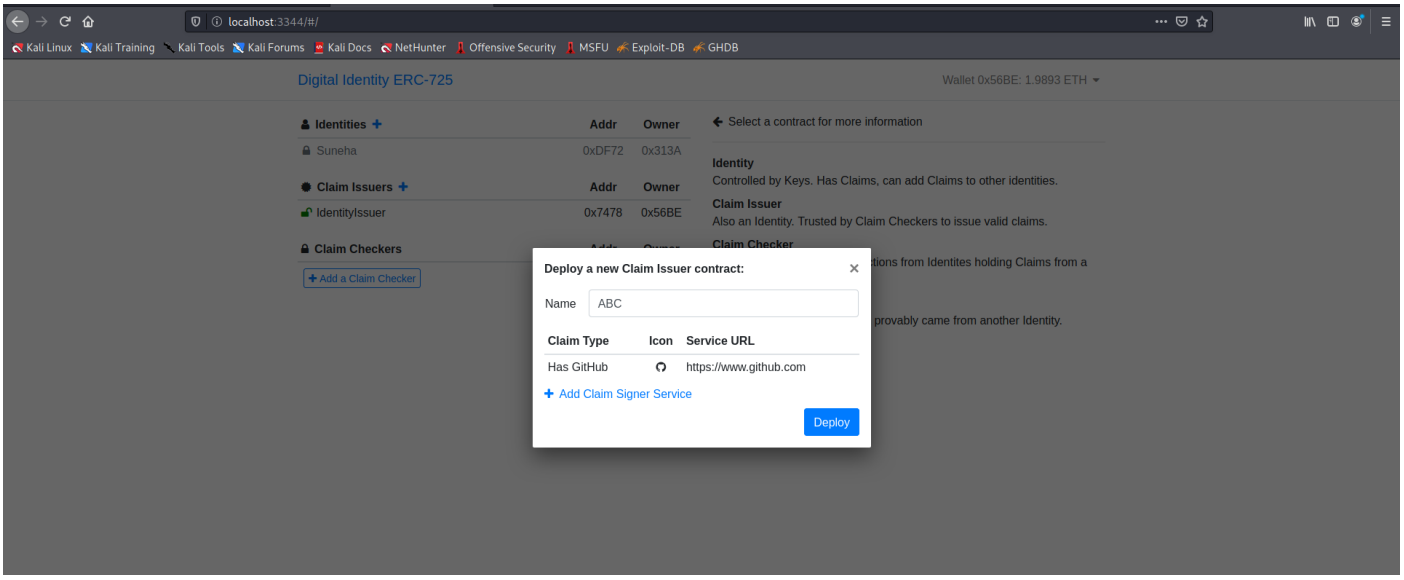


6. We then add claim signature to the issuer (here we are giving github as signature)

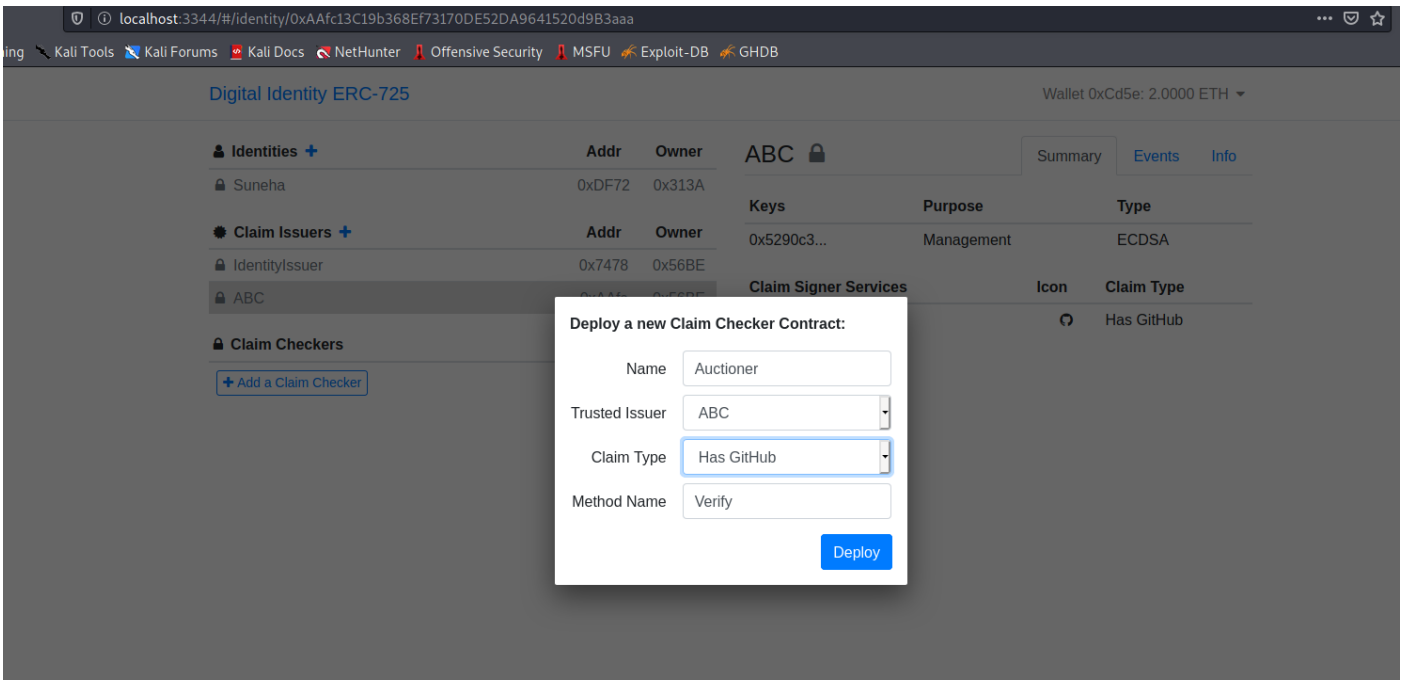


Here we have given ABC as the claim issuer

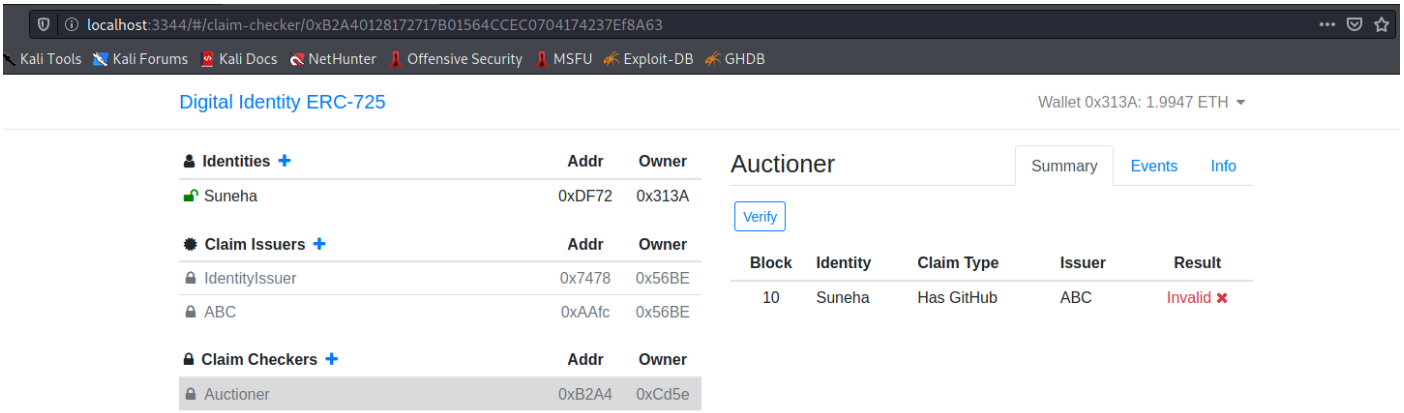




7. Again, change the Wallet to 0xCd5e and add a claim Checker i.e., the one who will verify the legitimacy of the claim made by the buyer. Here we give the checker's name as auctioner and the trusted 3rd party issuer as ABC who can verify whether the buyer has a GitHub account or not.

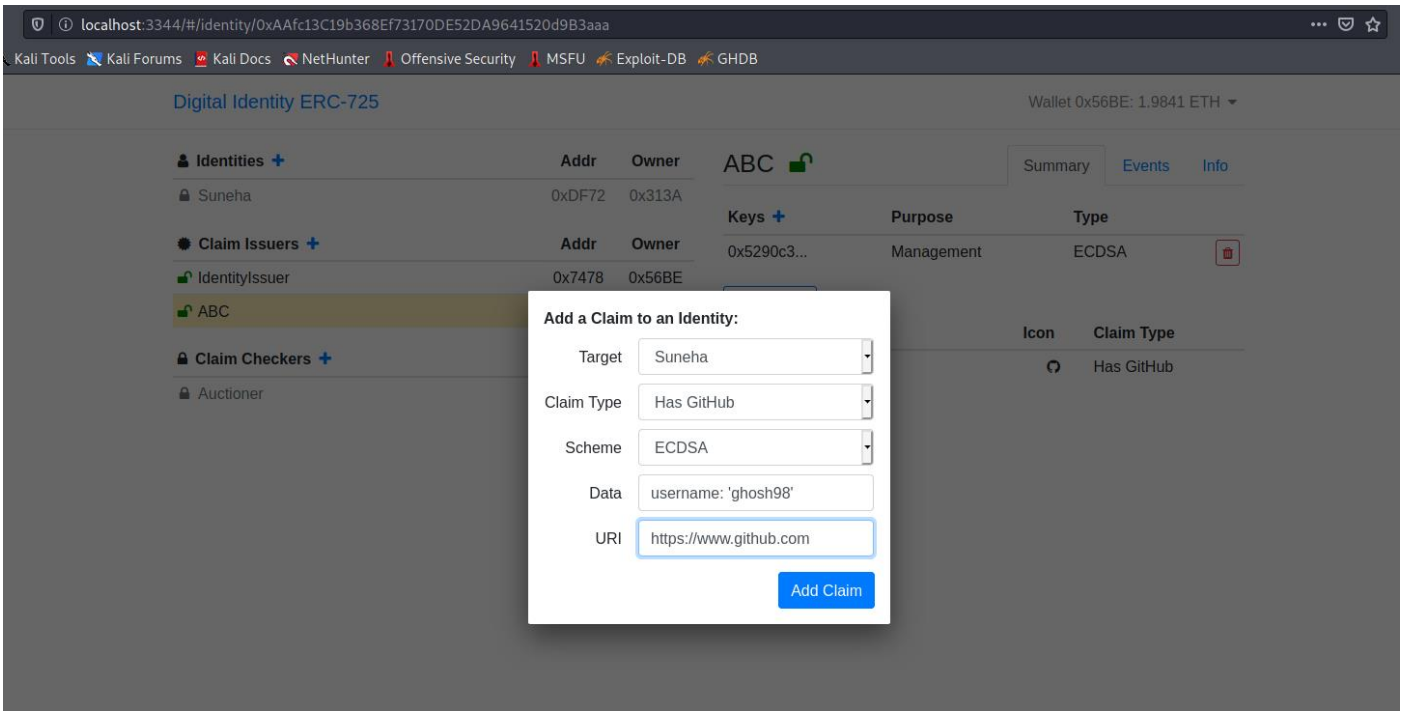


8. After this go to wallet id of the Identities and select Suneha
9. We have then added a self-claim of 'has GitHub' for the auctioner to see and verify.
10. Go to claim checker → Auctioner and click on verify. Due to the absence of the 3rd party claim issuer the self-claim of the buyer becomes invalid

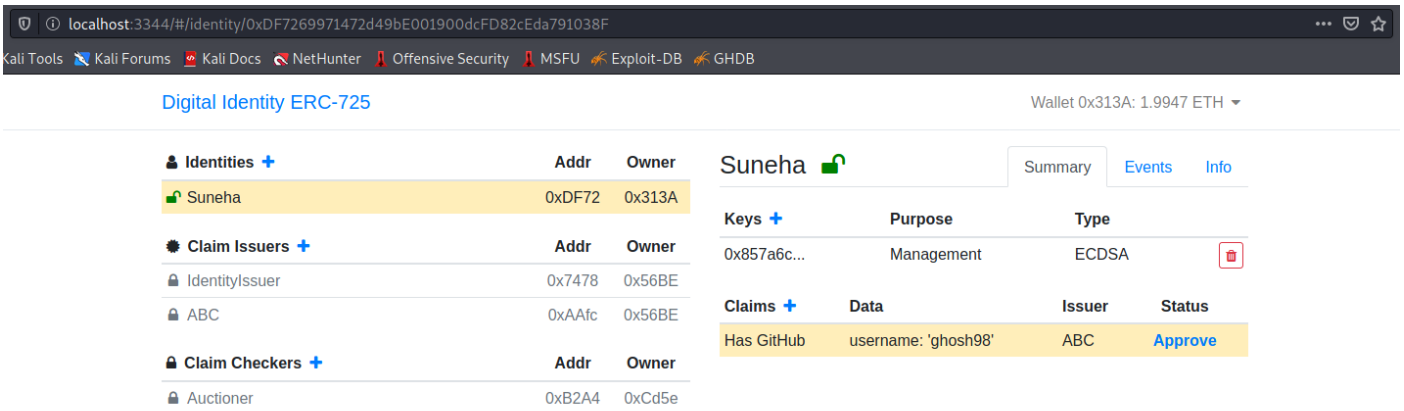


11. Then go to the 2nd wallet Id i.e. of the claim issuer and click on add a claim for an identity.

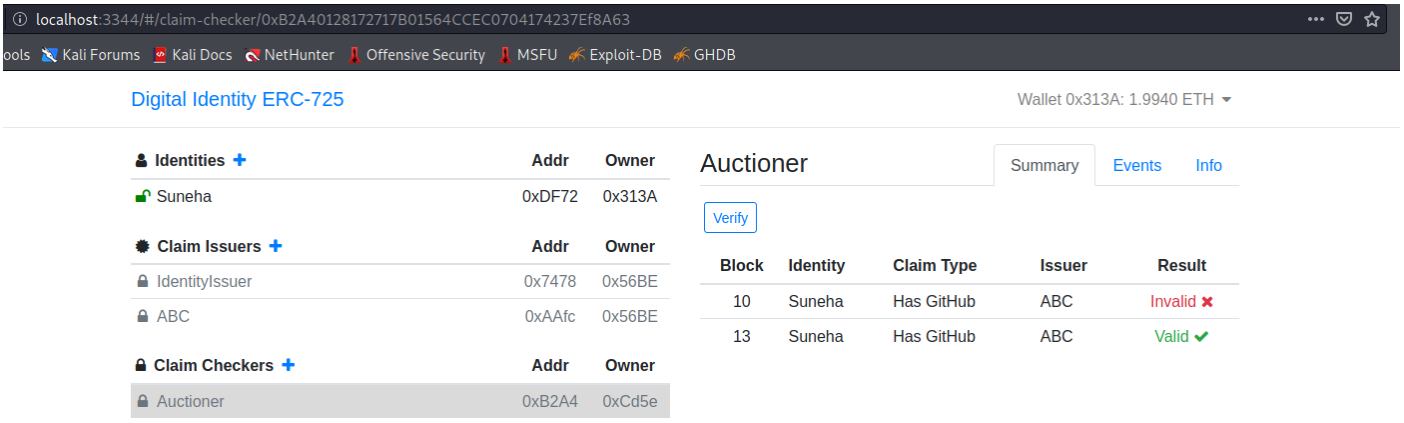
12. Select the identity (in this case suneha) and add a claim saying 'has GitHub'

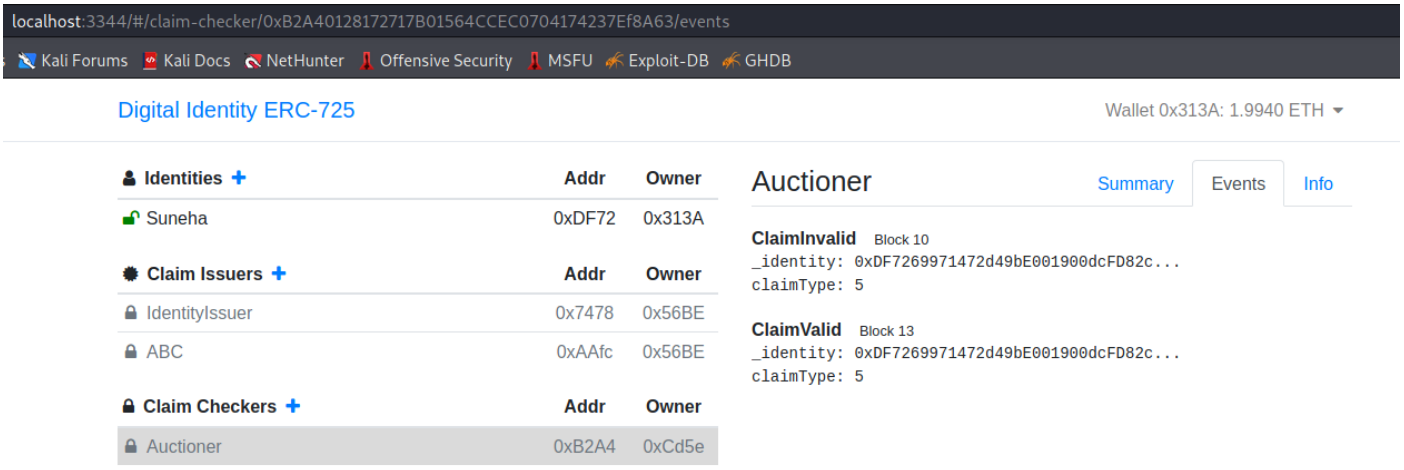


13. Now go to the 1st wallet id i.e. of the identities and approve the claim.



14. Now go to the claim checker → Auctioner and click on verify. This time the claim will be shown as valid since the claim has been issued by a trusted 3rd party of the claim checker.





The claimType is 5 because in the code we have given it as type 5.

15. After the claim getting valid, the auctioner can sell the property to the buyer i.e. transaction can proceed without any doubt.

The implementation of this project has been screen recorded with explanation, please check the google drive link below: -

https://drive.google.com/file/d/1Xnv4CPFIGjKzfRAwx4KP7qjdh-D5vzeF/view