Sawtooth Permissions Lab Manual

Pre-requisites

- Install Docker Engine and Docker Compose.
- Download the Sawtooth permissions folder from Moodle (cookiejar_permissions.zip).
- Extract the files

Step 1: Start the Sawtooth Docker Environment

To start up the environment, perform the following tasks:

- Open a terminal window.
- Change your working directory to the same directory which is extracted now and see the docker file. (docker-compose.yaml).
- Run the following command:

```
$ sudo docker-compose  up
```

Downloading the Docker images for the Sawtooth environment can take few minutes.

Step 2: Log Into The Validator Container

- Log into the validator container by opening a new terminal window and running the following command. Note that validator specifies the validator container name.

```
$ sudo docker exec -it validator bash
```

1. Introduction : Working with Cookie-jar

We are already used with the cookie-jar program and we know the BAKE, EAT and COUNT operations.
Note: To run the cookie-jar, we can use the UI hosted in http://localhost:3000.

1.1. In the validator container create two pairs of keys named Alice and Bob.

```
sawtooth keygen Bob
sawtooth keygen Alice
```

Note: The keygen command can also be used to create keys with specific names

1.2. Then display the keys using,

```
cat ~/.sawtooth/keys/Bob.priv
cat ~/.sawtooth/keys/Alice.priv
```

1.3. Copy the private keys one by one and login to use cookiejar application.

Note : You can bake, eat and count the cookies. Everything will be logically correct and working.

2. SETTING UP PERMISSIONS AND ROLES

The permissions are set by using Identity Transaction Family.

Note: Run the following commands from the **validator container**:

2.1. Add your public key to the list of allowed keys,

```
sawset proposal create --key ~/.sawtooth/keys/my_key.priv
sawtooth.identity.allowed_keys=$(cat
~/.sawtooth/keys/my_key.pub) --url http://rest-api:8008
```

Note: Here we are adding the public key 'my_key.pub' to the list of allowed keys to change settings

2.2. To check this setting use to following command

```
sawtooth settings list --url http://rest-api:8008 --format json
```

Note: You can see that your key is now added to the list of allowed keys.
sawtooth.identity.allowed_keys:
024b180ad91147f8e5ba33cc0815bf2b27003b7d0b935fb11cdfe a7d2972eac6d9

2.3. Once your public key is stored in the setting, use the command *sawtooth identity policy create* to set and update roles and policies.

For example, running the following command will create a policy that permits all and is named policy_1:

```
sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY *" --url
http://rest-api:8008
```

Note: Here policy_1 is the name of the policy file we are creating. We can use PERMIT_KEY to permit transactors and DENY_KEY to block specific transactors.

2.4. To view the policy enter the following command,

sawtooth identity policy list --url http://rest-api:8008 --format json

2.5. Use the command *sawtooth identity role create* to create a role for this policy.

The following example sets the role for *transactor* to the policy that permits all:

sawtooth identity role create --key ~/.sawtooth/keys/my_key.priv transactor policy_1 --url http://rest-api:8008

2.6. To view the role, enter the following command

sawtooth identity role list --url http://rest-api:8008 --format json

3. Giving permission to specific keys

3.1. Now we are going to give permission to any one of the two.

3.2. But before giving permissions to Bob or Alice you should give your public key (my_key) the permission to be a transactor.

3.2.1 For that you should obtain your public key using the following command.

cat ~/.sawtooth/keys/my_key.pub

3.2.2 Give permission by editing the following command,

sawtooth identity policy create --key ~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY my_key" --url http://rest-api:8008

Note : replace the highlighted part my_key with the public key obtained in step 3.2.1

And it will look something like

sawtooth identity policy create --key ~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY 024b180ad91147f8e5ba33cc0815bf2b27003b7d0b935fb11cdfea7d2972eac6d9" --url http://rest-api:8008

3.3. Now we can give permission to Alice by editing the policy command again by adding the keys of our choice.

3.3.1 For giving permission to Alice, we should again obtain the public key of Alice using the following command.

cat ~/.sawtooth/keys/Alice.pub

3.3.2 Now use the obtained public key and permit Alice as a transactor.

sawtooth identity policy create --key ~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY my_ key" “PERMIT_KEY key_of_Alice” --url http://rest-api:8008

NOTE : replace the highlighted part with the public key of Alice which is obtained in step 3.2.1. And as we have done already, my_key should be also be replaced with the key in step 3.2.1

And this will look something like

```
sawtooth identity policy create --key
~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY
024b180ad91147f8e5ba33cc0815bf2b27003b7d0b935fb11
cdfea7d2972eac6d9" "PERMIT_KEY
0299769d6b2fb2e59932d48bcc2f534459a4b5a59171b79fe
c2da2a884b1c81372" --url http://rest-api:8008
```

3.4. You can check your policy now,

```
sawtooth identity policy list --url http://rest-api:8008 --format json
```

3.5. Now go to the http://localhost:3000 and try to login with the Bob's private key again or any newly generated key and try out all the functionalities.
NOTE : Here we are submitting the transaction using the keys which are not permitted

3.6. If everything is done right, this transactions will not be executed.
NODE : The count option will show as undefined/unchanged.

3.7. Now login with the private key of Alice. See all the functionalities, and you can see, it works normal.

3.8. You can try creating new keys in the validator bash and can try adding more permitted keys by going to the validator and editing the PERMIT_KEY code.

For example,
sawtooth identity policy create --key ~/.sawtooth/keys/my_key.priv policy_1 "PERMIT_KEY 024b180ad91147f8e5ba33cc0815bf2b27003b7d0b935fb11cdfea7d2972eac6d9" "PERMIT_KEY 0299769d6b2fb2e59932d48bcc2f534459a4b5a59171b79fec2da2a884b1c81372" “DENY_KEY 01759ghy642fb2e59932d48bcc2f534459ub4m59171b79gbu2da2a884b1c51247” --url http://rest-api:8008

*Note**: Notice that we are creating separate entry for each key. If this is not done, only the permitted keys will be able to submit transactions*

4. Additional Exercise : Assigning a different role

4.1. We can also specify different roles.

For that we create a new policy, Policy_2 with the key of Bob. (Don’t forget about your public key)

sawtooth identity policy create --key ~/.sawtooth/keys/my_key.priv policy_2 "PERMIT_KEY my_ key" “PERMIT_KEY key_of_Bob” --url http://rest-api:8008

4.2. We can use the following command to assign a policy to a particular transaction processor.

sawtooth identity role create --key ~/.sawtooth/keys/my_key.priv transactor.transaction_signer.cookiejar policy_2 --url http://rest-api:8008

Note: After applying this command, if a transaction is received for 'cookiejar' transaction family that is signed by a transactor who is not permitted by the policy, the batch containing the transaction will be dropped.

4.3. Different combinations of PERMIT_KEY and DENY_KEY can be tried.
Allow all the transactors but deny only a particular key.

.