ASSIGNMENT 5

AUTHORIZATION

Neilos Kotsiopoulos

neko@itu.dk

1) Problem 1: ACL (Ambient Authority)

Alice can read and write to the file a, read the file b, and execute the file c. bob can read a, read and write to b, and has no access to c.

Part 1: Write access control lists for this situation

ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects

- File a:
  <alice, rw-> , <bob, r-->
- File b:
  <alice, r--> , <bob, rw->
- File c:
  <alice, --x> , <bob, --->

Where r for read, w for write, and x for execute

Part 2: Write capability lists for this situation

Capability lists specify the operations each user is allowed to perform

- alice:
  <a, rw-> , <b, r--> , <c,--x>
- bob:
  <a, r--> , <b, rw-> , <c,--->

Part 3: Say you need to, on the one hand, a) revoke all write permissions to a specific file, and on the other hand b) revoke all write permissions of a specific user. In these two scenarios, what is the difference between access control lists and capability lists, in terms of what you need to do to achieve the desired effect?

a) Revoke -w on specific file?

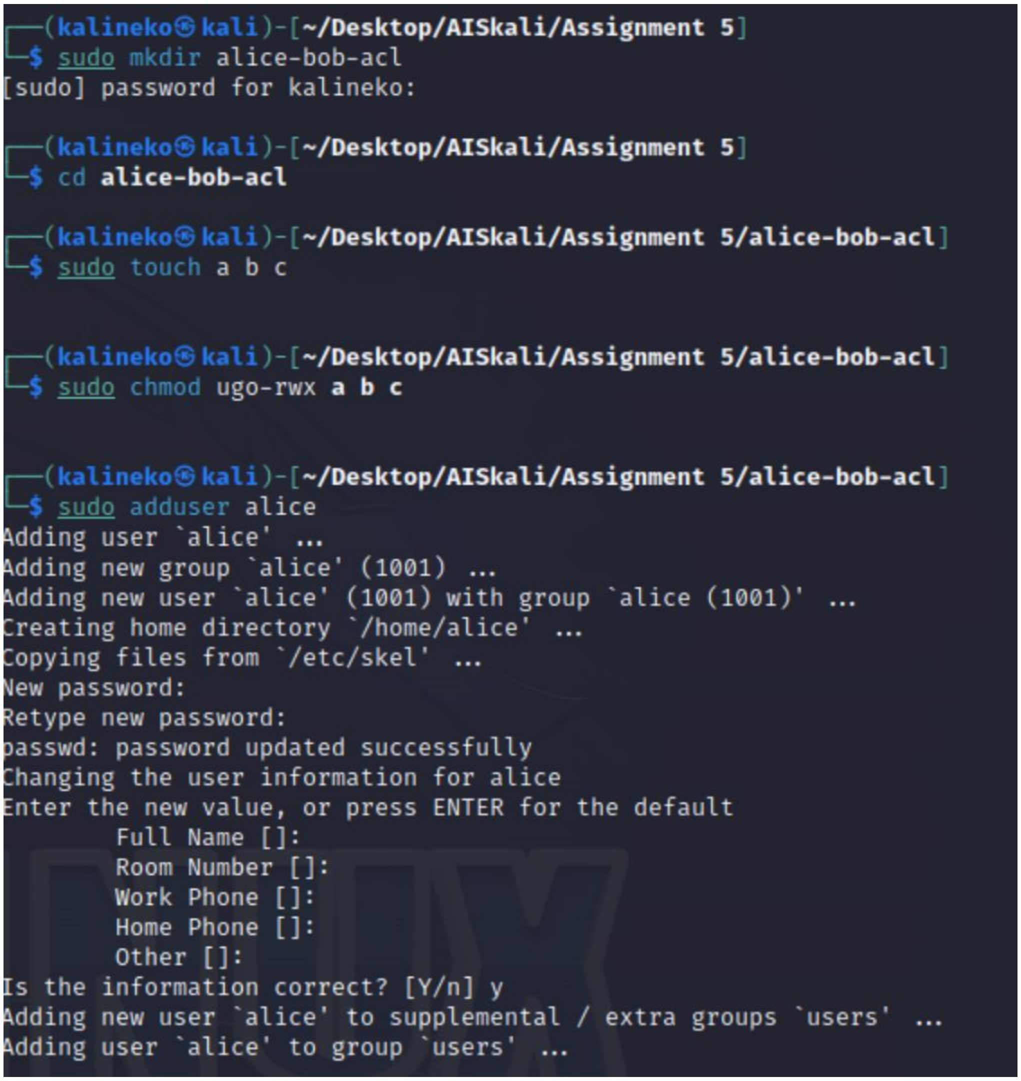
- With ACLs, I have to go to the specific file's ACL and remove the write permissions from all users listed there, centralized authority control
- With CLs, I need to visit each user's capability list and remove the write permission for that specific file, decentralized access control and complex (if the number of users is high)

b) Revoke -w on specific user?

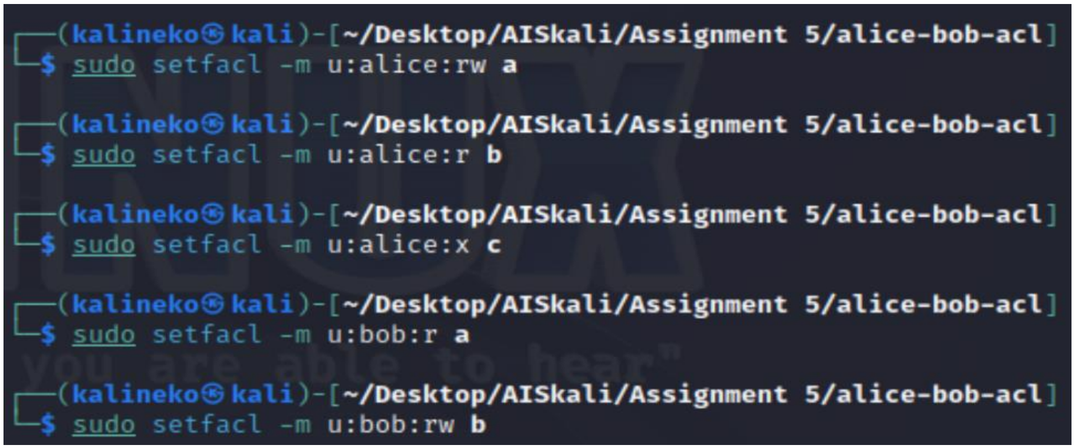
- With ACLs, I need to visit the ACL of each file and check if the user is listed there. If they are, I need to remove their write permission, which is time consuming
- With CLs, I just need visit the specific user's capability list and remove the write permissions from all files listed there, faster approach

Part 4: As root, create a directory alice-bob-acl, create the above files a, b, and c, and remove all permissions for everyone from these files. Then, add the above described permissions into the access control lists of the files. tar the result (see commands below), and include the tarball in your submission:

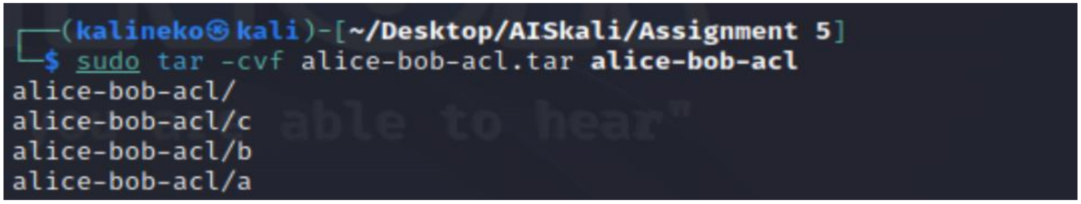
Please see the tar file included in my submission. Here's some snippets from the process following the assignment's instructions



```
┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5]
└─$ sudo mkdir alice-bob-acl
[sudo] password for kalineko:

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5]
└─$ cd alice-bob-acl

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo touch a b c


┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo chmod ugo-rwx a b c


┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo adduser alice
Adding user `alice' ...
Adding new group `alice' (1001) ...
Adding new user `alice' (1001) with group `alice (1001)' ...
Creating home directory `/home/alice' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alice
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding new user `alice' to supplemental / extra groups `users' ...
Adding user `alice' to group `users' ...
```



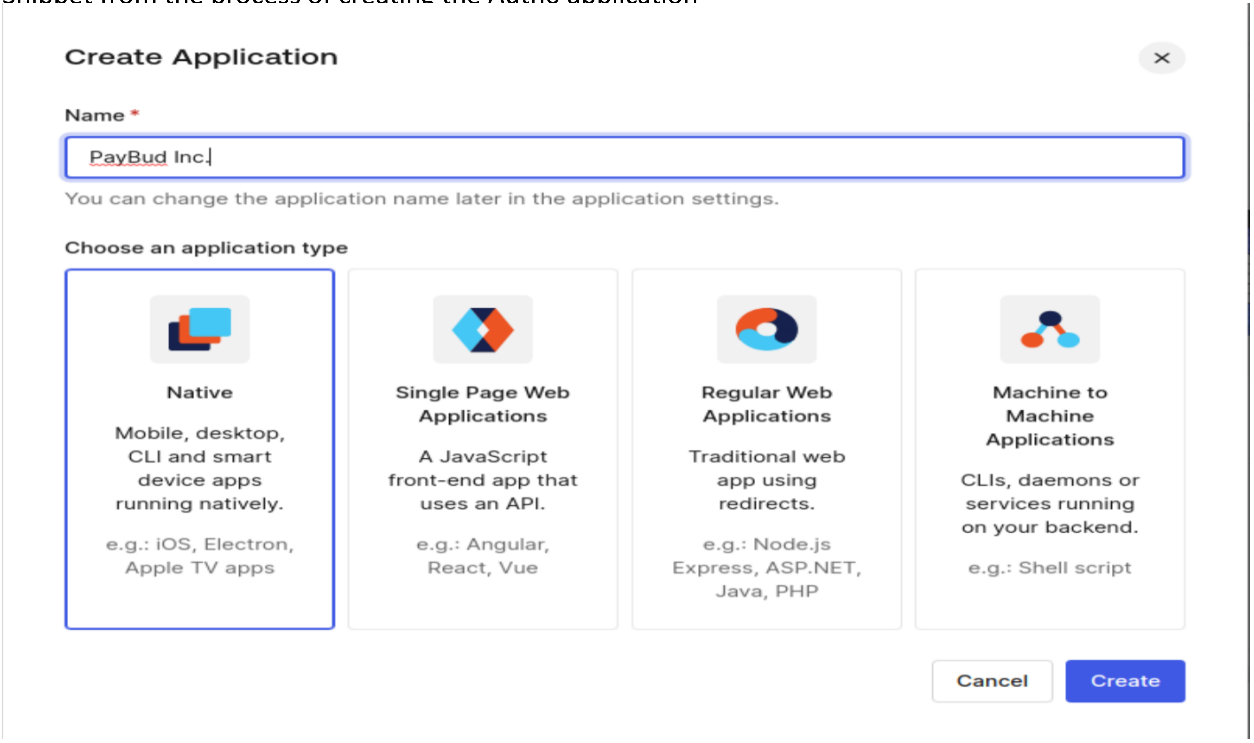
```
┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo setfacl -m u:alice:rw a

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo setfacl -m u:alice:r b

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo setfacl -m u:alice:x c

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo setfacl -m u:bob:r a

┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5/alice-bob-acl]
└─$ sudo setfacl -m u:bob:rw b
```



```
┌──(kalineko㉿kali)-[~/Desktop/AISkali/Assignment 5]
└─$ sudo tar -cvf alice-bob-acl.tar alice-bob-acl
alice-bob-acl/
alice-bob-acl/c
alice-bob-acl/b
alice-bob-acl/a
```

2) Problem 2: DAC (Authentication in Microservices)

Snippet from the process of creating the Auth0 application



Create Application

Name *

PayBud Inc.

You can change the application name later in the application settings.

Choose an application type

Native

Mobile, desktop, CLI and smart device apps running natively.

e.g.: iOS, Electron, Apple TV apps

Single Page Web Applications

A JavaScript front-end app that uses an API.

e.g.: Angular, React, Vue

Regular Web Applications

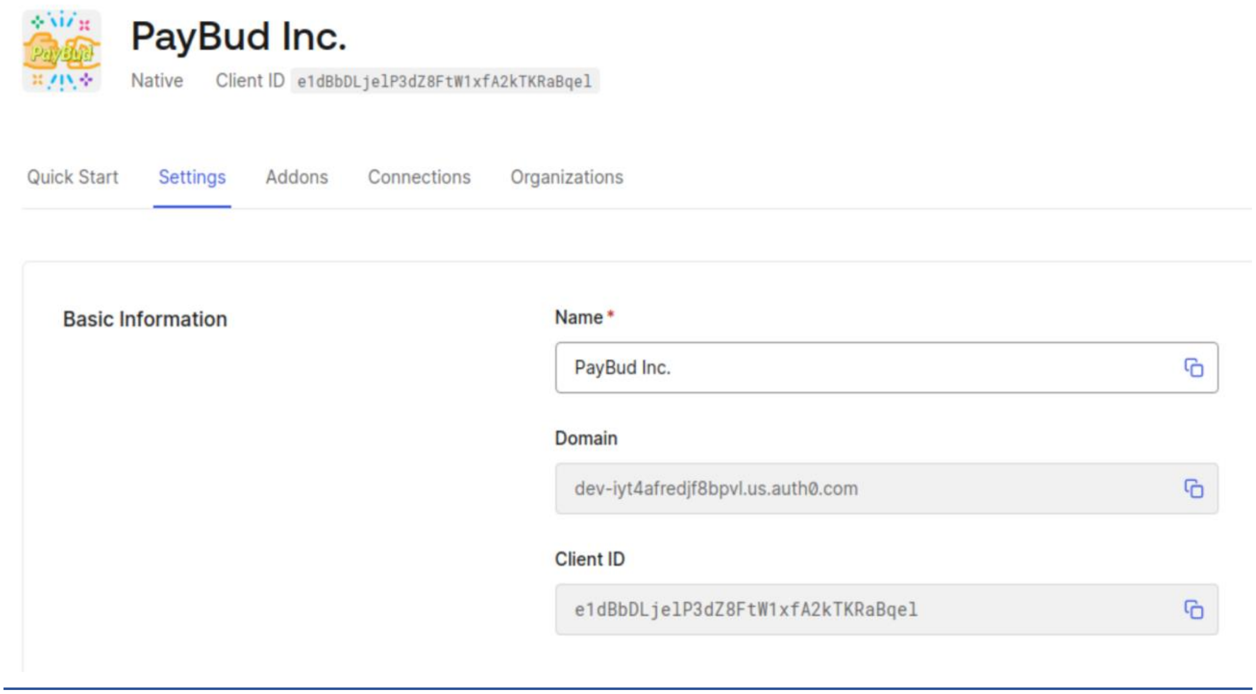
Traditional web app using redirects.

e.g.: Node.js Express, ASP.NET, Java, PHP

Machine to Machine Applications

CLIs, daemons or services running on your backend.

e.g.: Shell script



PayBud Inc.

Native Client ID e1dBbDLjelP3dZ8FtW1xfA2kTKRaBqel

Basic Information

Name *

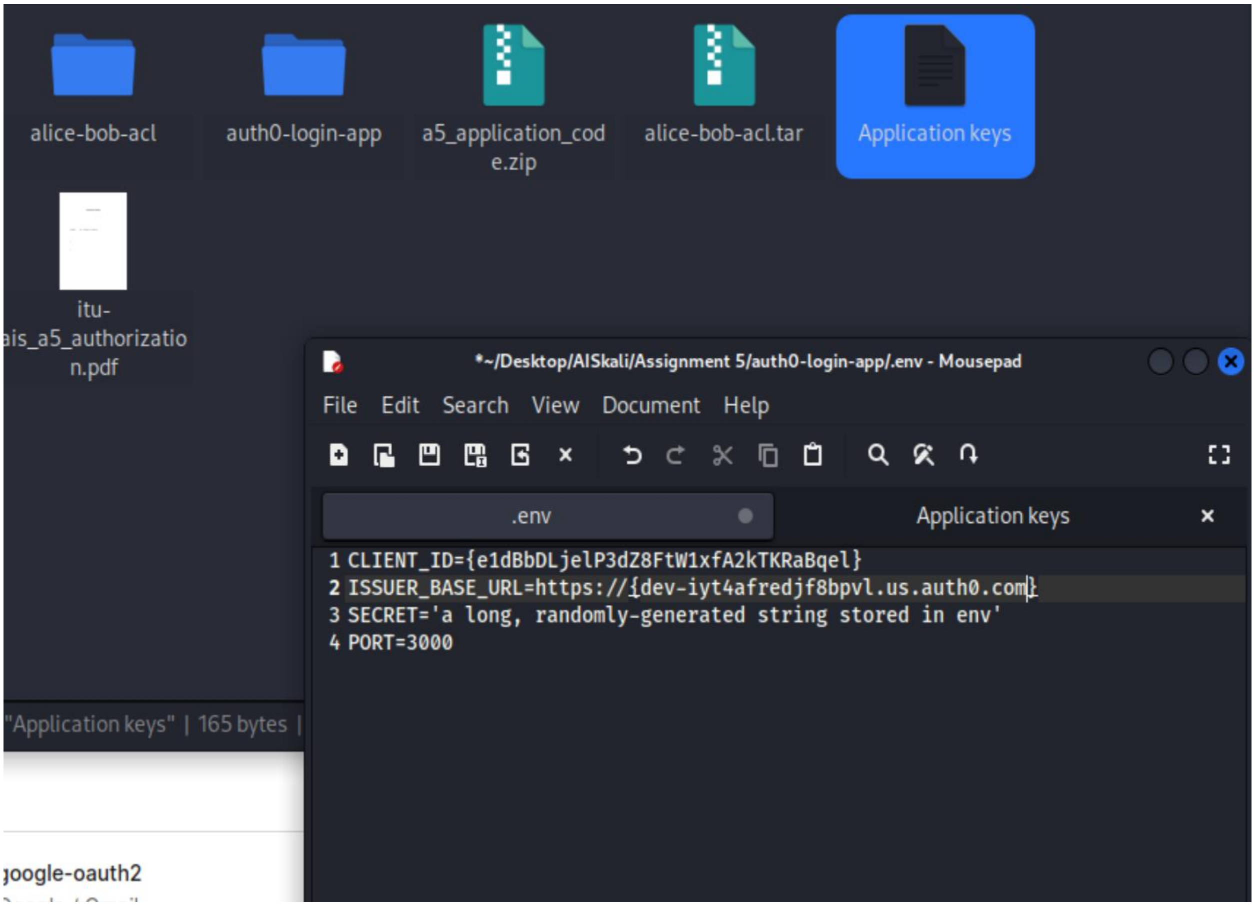
PayBud Inc.

Domain

dev-iyt4afredjf8bpvl.us.auth0.com

Client ID

e1dBbDLjelP3dZ8FtW1xfA2kTKRaBqel



```
CLIENT_ID={e1dBbDLjelP3dZ8FtW1xfA2kTKRaBqel}
ISSUER_BASE_URL=https://{dev-iyt4afredjf8bpvl.us.auth0.com}
SECRET='a long, randomly-generated string stored in env'
PORT=3000
```

Part 1: Who or What is the 1) resource owner, 2) client, 3) authorization server and 4) resource server?

1. Resource Owner

   The Resource owner is the user, in our case aliceX@mailinator.com

2. Client

   The Client is the Paybud Inc. application on http://localhost:3000

3. Authorization server

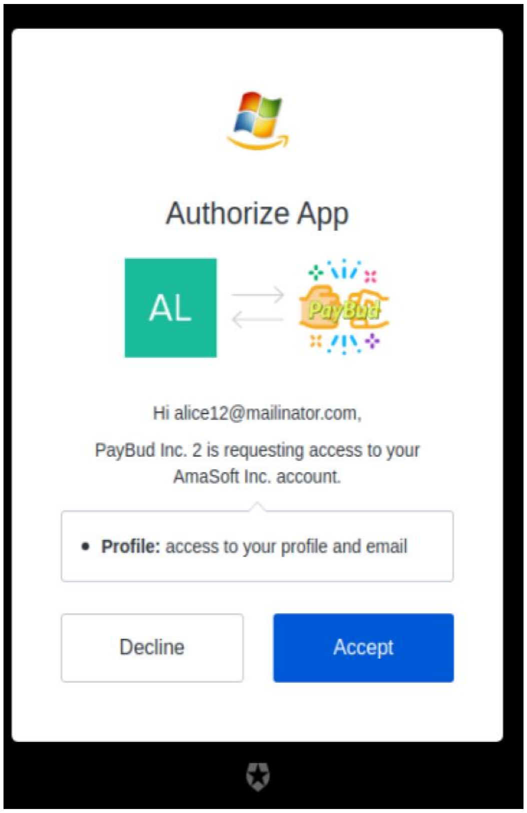
   The AS is Auth0

4. Resource Server

   AmaSoft Inc. is the Resource Server in our example (as setted in the account settings)

Part 2: Create a new user account for the chosen e-mail address (Login → Sign up). Then log into that account. Explain what happened when you logged in; what sends what message to what? (Include the names of the entities)

Once I have created an account by providing my email < alice12@mailinator.com> and password , the client has redirected me to the authorization server (Auth0) that verified this credentials and then showed my the consent page that requests access to my AmaSoft Inc. account

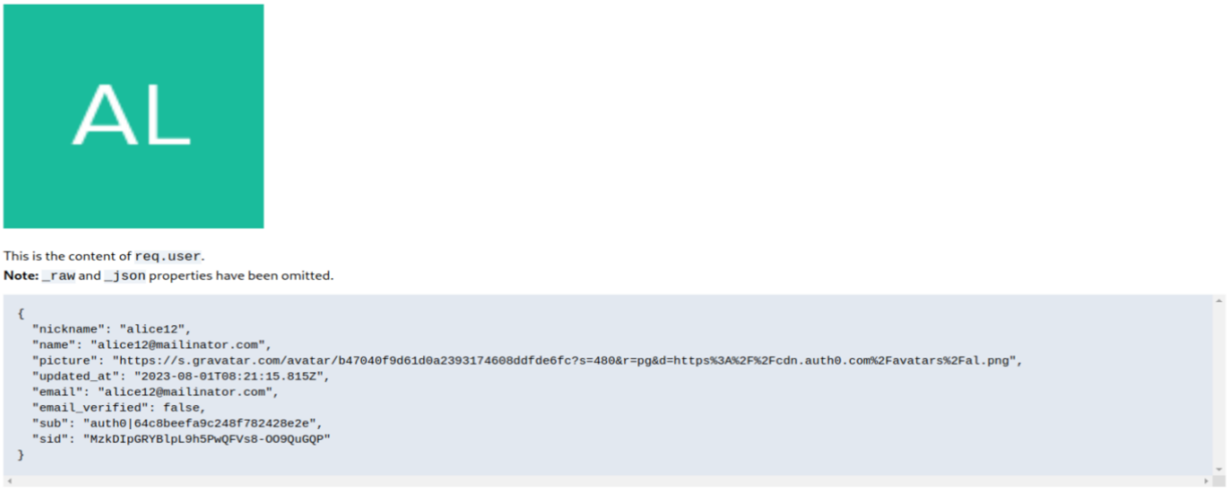

Authorize App

Hi alice12@mailinator.com,

PayBud Inc. 2 is requesting access to your AmaSoft Inc. account.

- **Profile:** access to your profile and email

Once I log in, I have already the client to authorize my data and the authorization server redirects me back to the client (including the authoriation code). The client exchanges the authorization code for an access token by making a POST request to the authorization server.



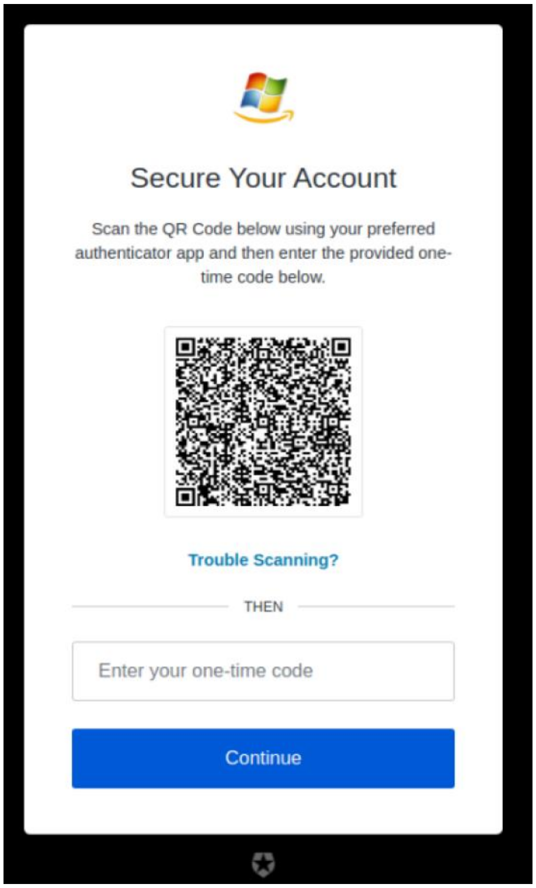
This is the content of `req.user`.
Note: `_raw` and `_json` properties have been omitted.

```
{
  "nickname": "alice12",
  "name": "alice12@mailinator.com",
  "picture": "https://s.gravatar.com/avatar/b47040f9d61d0a2393174608ddfde6fc?s=480&r=pg&d=https%3A%2F%2Fcdn.auth0.com%2Favatars%2Fal.png",
  "updated_at": "2023-08-01T08:21:15.815Z",
  "email": "alice12@mailinator.com",
  "email_verified": false,
  "sub": "auth0|64c8beefa9c248f782428e2e",
  "sid": "MzkDIpGRYBlpL9h5PwQFVs8-OO9QuGQP"
}
```

The authorization server authenticates the client and validates the authorization code. If everything checks out, it responds with an access token. The client uses the access token to make API requests to the resource server (AmaSoft Inc). The resource server validates the access token, and if it's valid, returns the requested data and the Client can display the User's information.

Part 3: Enable MFA in the identity provider (Security → Multi-factor Auth), by enabling "one-time password" and Require Multi-factor Auth: Always. Log out and in again to add a 2nd factor (e.g. Microsoft Authenticator). Then log out and in again. Explain which steps are added & where in the protocol run from Part 2

Once I enabled the Multi-Factor Authentication and logged out, then in the login screen it was presented a QR code in order to provide the one-time code from the Microsoft Auth app



Secure Your Account

Scan the QR Code below using your preferred authenticator app and then enter the provided one-time code below.

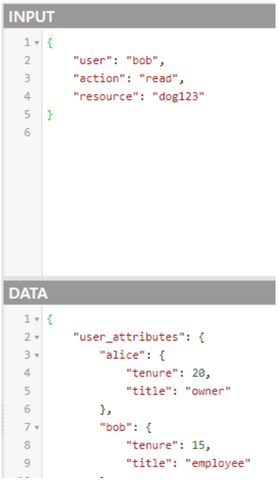
Once I provided the code, I managed to log in to the client. The steps that added are:

- Client Requests access token from authorization server
- Authorization server sends one-time-code to user
- User provides one-time-code to authorization server
- Authorization server sends access token to client application

3) Problem 4: MAC

Part 1: In POLICY, What does the following evaluate to?

1. Input.action
   evaluates to "read" action
2. data.user_attributes[input.user]
   evaluates to the attributes of the user bob, aka tenure = 15 and title = “employee”
3. data.pet_attributes[input.recource]
   evaluates to the attributes for the pet “dog123”, adopted = true, age = 2, breed = “terrier” and name = Toto

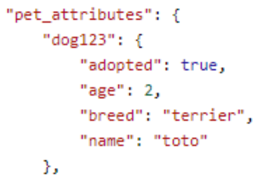


INPUT

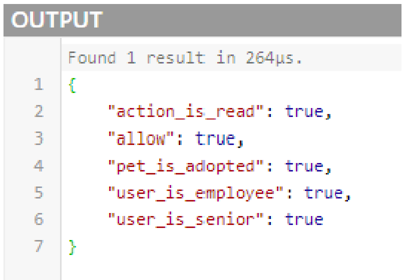
```
{
    "user": "bob",
    "action": "read",
    "resource": "dog123"
}
```

DATA

```
{
    "user_attributes": {
        "alice": {
            "tenure": 20,
            "title": "owner"
        },
        "bob": {
            "tenure": 15,
            "title": "employee"
```



```
"pet_attributes": {
    "dog123": {
        "adopted": true,
        "age": 2,
        "breed": "terrier",
        "name": "toto"
    },
```

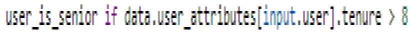


OUTPUT

```
Found 1 result in 264µs.
{
    "action_is_read": true,
    "allow": true,
    "pet_is_adopted": true,
    "user_is_employee": true,
    "user_is_senior": true
}
```

Part 2: The OUTPUT of a policy evaluation is a record with up to 8 attributes. Justify your answer to the following by referencing POLICY, INPUT and DATA

1. what is the (data-)type of all 8 possible attributes?
   All 8 possible attributes in the output record are Boolean types
2. when is action_is_{read, update} true?
   When the 'action' field in the INPUT is "read" or “update” accordingly
3. when is user_is_{owner, employee, customer} true?
   When the 'title' field for the specified user in the DATA is "owner" or “employee” or “customer” accordingly
4. when is user_is_senior true?
   When the 'tenure' field for the specified user in the DATA is greater than 8



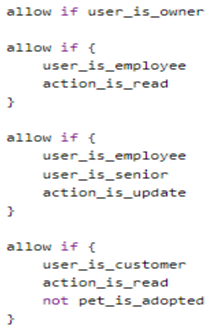
```
user_is_senior if data.user_attributes[input.user].tenure > 8
```

5. when is pet_is_adopted true?
   When the 'adopted' field for the specified resource in the DATA is true

6. when is allow true?

In the following cases:

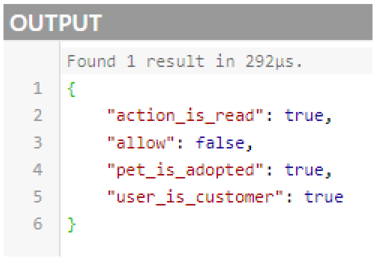
- if ‘user_is_owner’ = true
- if ‘user_is_employee’ AND ‘action_is_read’ = true
- if ‘user_is_employee’ AND ‘user_is_senior’ AND ‘action_is_update’ = true
- if ‘user_is_customer’ AND ‘action_is_read’ AND ‘pet_is_adopted’ = false



```
allow if user_is_owner

allow if {
    user_is_employee
    action_is_read
}

allow if {
    user_is_employee
    user_is_senior
    action_is_update
}

allow if {
    user_is_customer
    action_is_read
    not pet_is_adopted
}
```

Part 3: We will ask for the value of allow in the following. To justify your answer, trace the value of allow, i.e. “allow is true if X is true, which is true if Y is true, ...” (you can replace “true if X is true” by “implied by X”)

1. Consider the default INPUT
   What is the value of allow?
   As stated above allow is true if ‘user_is_employee’ AND ‘action_is_read’ = true. Bob is an employee, and the action is read, thus allow is true
2. Change the user attribute in the INPUT to alice
   What is the value of allow?
   Again, allow is true if ‘user_is_owner’ = true. Alice has the title “owner”, thus allow is true
3. Change the user attribute in the INPUT to dave
   What is the value of allow?
   Allow is false as dave does not require any of the afore mentioned criteria. Dave is a customer, the action is “read” but the attribute ‘pet_is_afopted’ is true, while it had to be false, in order to the allow attribute being true

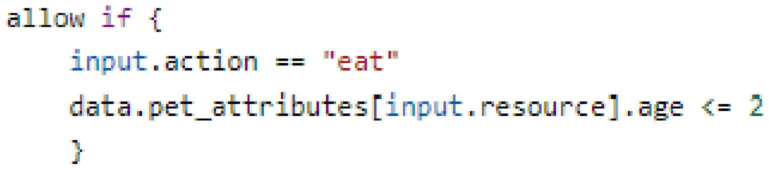


OUTPUT

```
Found 1 result in 292µs.
{
    "action_is_read": true,
    "allow": false,
    "pet_is_adopted": true,
    "user_is_customer": true
}
```

Part 4: Change the policy such that any user is allowed the action eat of animals that are less than or equal to 2 years of age

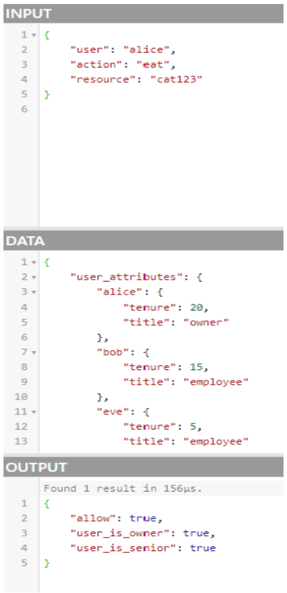
First, I added the following line to the polict to create the action “eat”



```
allow if {
    input.action == "eat"
    data.pet_attributes[input.resource].age <= 2
    }
```

1. Give example INPUT with action eat where allow is true

   Alice – eat – cat123



INPUT

```
{
    "user": "alice",
    "action": "eat",
    "resource": "cat123"
}
```

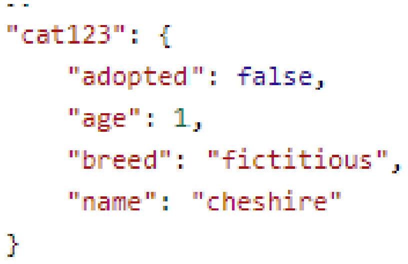
DATA

```
{
    "user_attributes": {
        "alice": {
            "tenure": 20,
            "title": "owner"
        },
        "bob": {
            "tenure": 15,
            "title": "employee"
        },
        "eve": {
            "tenure": 5,
            "title": "employee"
```

OUTPUT

```
Found 1 result in 156µs.
{
    "allow": true,
    "user_is_owner": true,
    "user_is_senior": true
}
```

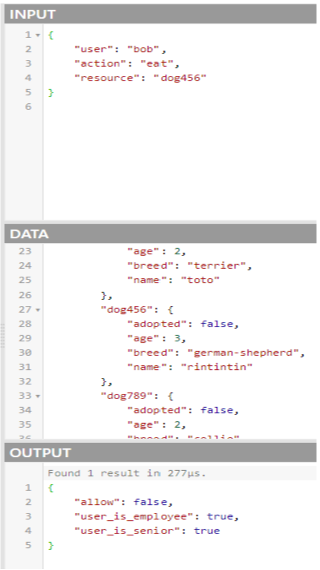
Allow is true as “cat123” is only 1 year old < 2 years old



```
"cat123": {
    "adopted": false,
    "age": 1,
    "breed": "fictitious",
    "name": "cheshire"
}
```

2. Give example INPUT with action eat where allow is false

   Bob – eat – dog456



INPUT

```
{
    "user": "bob",
    "action": "eat",
    "resource": "dog456"
}
```

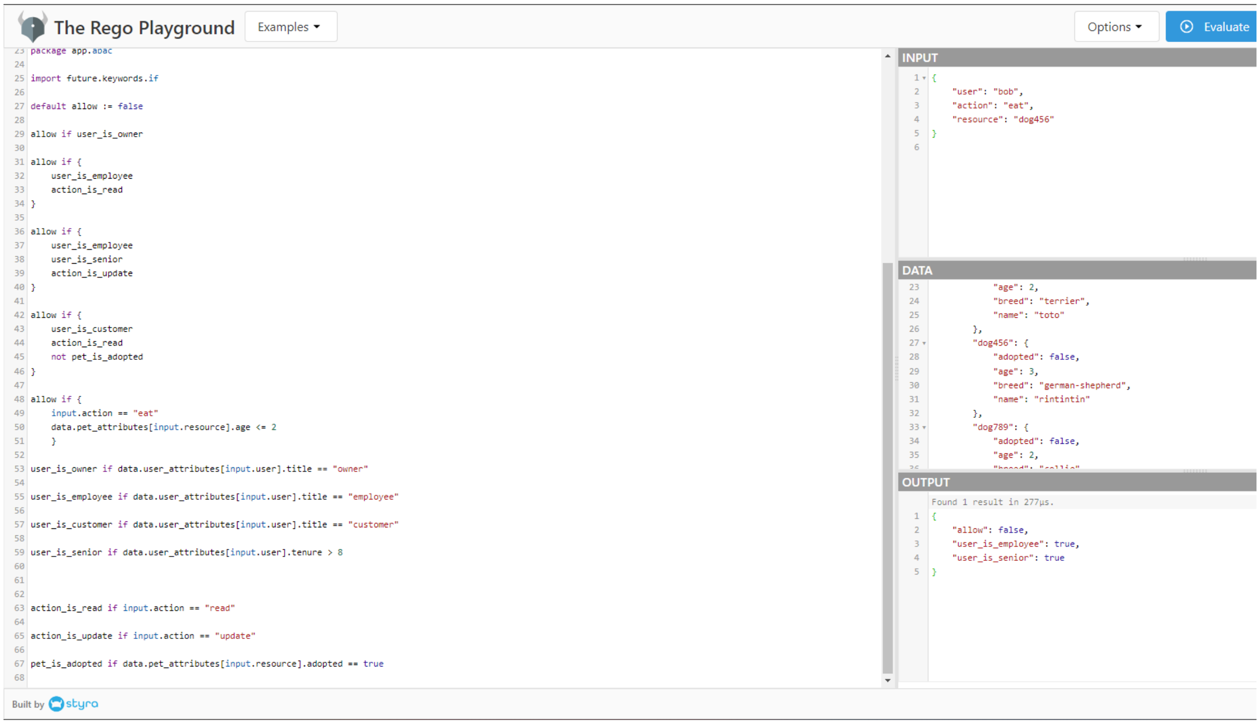
DATA

```
            "age": 2,
            "breed": "terrier",
            "name": "toto"
        },
        "dog456": {
            "adopted": false,
            "age": 3,
            "breed": "german-shepherd",
            "name": "rintintin"
        },
        "dog789": {
            "adopted": false,
            "age": 2,
            "breed": "collie"
```

OUTPUT

```
Found 1 result in 277µs.
{
    "allow": false,
    "user_is_employee": true,
    "user_is_senior": true
}
```

Allow is true as “dog456” is 3 years old > 2 years old



```
package app.abac

import future.keywords.if

default allow := false

allow if user_is_owner

allow if {
    user_is_employee
    action_is_read
}

allow if {
    user_is_employee
    user_is_senior
    action_is_update
}

allow if {
    user_is_customer
    action_is_read
    not pet_is_adopted
}

allow if {
    input.action == "eat"
    data.pet_attributes[input.resource].age <= 2
    }

user_is_owner if data.user_attributes[input.user].title == "owner"

user_is_employee if data.user_attributes[input.user].title == "employee"

user_is_customer if data.user_attributes[input.user].title == "customer"

user_is_senior if data.user_attributes[input.user].tenure > 8



action_is_read if input.action == "read"

action_is_update if input.action == "update"

pet_is_adopted if data.pet_attributes[input.resource].adopted == true
```

INPUT

```
{
    "user": "bob",
    "action": "eat",
    "resource": "dog456"
}
```

DATA

```
            "age": 2,
            "breed": "terrier",
            "name": "toto"
        },
        "dog456": {
            "adopted": false,
            "age": 3,
            "breed": "german-shepherd",
            "name": "rintintin"
        },
        "dog789": {
            "adopted": false,
            "age": 2,
            "breed": "collie"
```

OUTPUT

```
Found 1 result in 277µs.
{
    "allow": false,
    "user_is_employee": true,
    "user_is_senior": true
}
```