

Proiect de Programare – C++

-Sistem 0-trust security-

Student 1: Mitoiu Bogdan-Petru

Student 2: Turculeț Alexandru

I. Enunț

Studentul 1 se ocupă de aplicația „client”

- poate cere acces la aplicația „auth-server” pentru anumite date
- vede cererile și starea lor (în așteptare, acceptate, respinse)
- afișează/modifică datele la care a primit accept.

Studentul 2 se ocupă de aplicația „server”

- gestionează permisiunile unui utilizator la un anumit fișier
- poate crea și șterge utilizatori
- aprobă sau respinge cereri de acces

II. Structura datelor folosite de echipă

Vom folosi clasele:

- Fișier: nume, tip
- Utilizator: id: String, nume: String, Cereri: Cerere[]
- Cerere: id: String, titlu: String, fișier: Fișier, motivare: String, drept_cerut: String, id_soluționare: String
- RăspunsCerere: id: String, id_cerere: String, stream: (i/o)stream?, acceptat: bool, expirare: Date

III. Structura fișierelor

Vom folosi următoarele fișiere:

- files.json
 - vom stoca fișierele disponibile cu drepturile de acces per utilizator
 - format json

```
{  
  
  • "version": 1,  
  • "files": [  
  •   {  
  •     "name": "file1.txt",  
  •     "permissions": [  
  •       {  
  •         "username": "patrocle",  
  •         "permissions": "rw"  
  •       }  
  •     ],  
  •     "type": "text"  
  •   },  
  •   • ș.a.m.d.  
  • ]  
  • }
```
- user.manifest
 - format json
 - {

 • "version": 1,
 • "id": "cc45d28f-8eca-4be4-ae3c-f6646357cbfa",
 • "name": "Alexandru Turculeț",
 • "requests": [
 • {
 • "id": "04d992d8-be9e-4278-85cc-293665cde0f1",
 • "title": "Cerere acces pentru file1.txt",
 • "requested_perms": "rw",

- "body": "Lorem ipsum dolor sit amet",
- "response_id": "ceb52253-da46-4474-b6f1-579955e0edf3"
- }
-],
- "responses":[
- {
- "id":"ceb52253-da46-4474-b6f1-579955e0edf3",
- "request_id": "04d992d8-be9e-4278-85cc-293665cde0f1",
- "answer": "accepted"
- }
-]
- }

Comenzi

Client:

- zerotrust-get auth <user> # autentifică clientul cu un utilizator
- zerotrust-get list # listează fișierele disponibile
- zerotrust-get read <file> # citește un fișier (dacă are permisiunea)
- zerotrust-get copy <file> #copiază un fișier (dacă are permisiunea de citire)
- zerotrust-get set <file> <source> # suprascrie un fișier prin conținutul <source> (dacă are permisiunea)
- zerotrust-get append <file> [--file <source>] <text> #adaugă la un fișier
- zerotrust-get deauth # deautentifică
- zerotrust-get request <file> --perms <permissions> --title <title> --body <body> #cere permisiunea pentru un fișier
- zerotrust-get request <file> --perms <permissions> --title <title> --body-file #la fel ca mai sus, conținutul cererii fiind citit dintr-un fișier
- zerotrust-get clear-perms <file>

Server

- Zerotrust-admin list-users # listează utilizatorii
- zerotrust-admin list-files # listează fișierele
- zerotrust-admin add-user <name> # adaugă un utilizator

- `zerotrust-admin delete-user <name> # șterge un utilizator`
- `zerotrust-admin index # indexează fișierele`
- `zerotrust-admin list-permissions <file> [--user <user>] # afișează permisiunile pentru un fișier (evtl. pt un utilizator)`
- `zerotrust-admin grant --user <user> --file <file> <permission> # acordă permisiunea unui utilizator`
- `zerotrust-admin revoke user <user> --file <file> <permission> # revocă permisiunea unui utilizator`
- `zerotrust-admin revoke-all <user> --file <file> # revocă toate permisiunile`
- `zerotrust-admin list-requests [--user <user>] # listează cererile (evtl. de la un utilizator)`
- `zerotrust-admin accept-request <id> # acceptă cererea cu id-ul dat`
- `zerotrust-admin deny-request <id> # respinge cererea cu id-ul dat`