

Raport - TEMA 2 - **Froxy**

Pricop Tudor-Constantin : 2A2

December 2021

Universitatea "Alexandru Ioan Cuza" Iasi / Facultatea de informatica

Abstract

Raportul din acest document contine detalii de structura si implementare a proiectului Froxy(A), realizat in limbajul de programare C, precum si portiuni de cod importante.

1 Introducere

Aceasta fisa are rolul de a prezenta detalii legate de constructia unui proiect din categoria (A), Froxy, proiect realizat in limbajul de programare C. Aplicatia este un exemplu de implementare a interfetei client-server, orientata pe protocolul FTP.

Legatura dintre client si serverul FTP de unde se preiau date este intermediata de un server proxy care actioneaza ca un filtru asupra datelor preluate, stabilind atat dimensiunea maxima si tipul fisierelor, cat si anumite politici de securitate: cine poate descarca informatie, de la ce gazda, in ce perioada de timp, etc. si stocandu-le intr-un folder temporar contruit pe masina ce ruleaza proxy-ul.

Un utilizator obisnuit va putea sa se conecteze la un server FTP, cu care va comunica prin intermediul proxy-ului, putand sa se plimbe printre directoarele acestuia, sa afiseze fisierele disponibile, sa descarce continutul acestora, etc. Un administrator va avea in plus dreptul de a crea alte conturi administrator, dar si dreptul de a modifica fisierul JSON de configurare a restrictiilor client \rightarrow server FTP.

2 Tehnologii utilizate

In acest proiect, pentru comunicarea client-proxy se va folosi protocolul TCP/IP in mod concurent, asigurandu-se astfel integritatea datelor transmise: in cadrul transferului de date este foarte important ca acestea sa ramana in aceeasi forma in care au fost transmise. Aceasta conditie nu putea fi respectata in cadrul protocolului UDP. Partea de concurenta este implementata cu ajutorul primitivei `fork()`, prin crearea de fii, fiecare responsabil de gestionarea request-urilor unui singur client. Pentru a nu bloca masina care ruleaza proxy-ul, acesta este limitat la 15 clienti simultani, alti clienti fiind rugati sa astepte eliberarea unui loc pana sa se poata conecta si ei.

De asemenea intra in joc si protocolul FTP(*RFC 959*) pentru comunicarea intre proxy si un server FTP online. Intre acestea comunicarea se realizeaza pe baza a doua canale: unul prin care se trimit comenzi(e.g. login, request de fisiere) si se primesc raspunsuri precedate de un cod care reprezinta fie raspuns pozitiv, fie incomplet, fie o eroare(codurile se gasesc in documentatie). Cel de-al doilea canal de comunicatie este destinat exclusiv transmiterii de fisiere/liste de date, fiind construit si distrus dupa un request completat.

Proxy-ul are rolul sa primeasca cereri de la clienti si sa le filtreze, parsand un fisier de configurare care contine diverse conditii(e.g. dimensiune maxima fisiere, host-uri de incredere). De asemenea, la pornire acesta isi construiesc un folder temporar in care sunt depozitate fisierele cerute de clienti, in cazul in care mai multi clienti vor sa downloadeze aceleasi fisiere se va face un singur request catre serverul FTP. La

finalul executiei proxy-ului(admin-ul da semnalul quit) acesta va sterge folderul temporar si continutul acestuia pentru a nu suprasolicita sistemul pe care este rulat. Gestionarea userilor este realizata intr-o baza de date, in care este retinut numele de utilizator, parola, tipul(obisnuit/admin) si daca acesta este logat momentan pe vreun client.

3 Arhitectura aplicatiei

Urmeaza sa fie prezentata arhitectura aplicatiei in scheme informale, acestea fiind impartite in segmente distincte deoarece la fiecare nivel in comunicatie sunt detalii diferite de functionare. Clientul si serverul proxy se pot afla pe dispozitive diferite, dar conectate la aceeasi retea de internet, folosindu-se de ip-ul distribuitorului pentru a putea comunica. De asemenea in memorie se afla si o baza de date care contine informatii despre utilizatori: id, username, password, type, astfel incat se poate face diferenta intre cine poate folosi anumite functionalitati. Doar admin-ul root(user:admin pass:admin) poate crea conturi de tip admin, cele de tip user obisnuit putand fi create de orice client.

Clientul:

Acesta se conecteaza la server, se logheaza ca prim pas, iar apoi initiaza un loop de comenzi de interogare si preluare de date de pe proxy. Cand isi termina ciclul de viata trimite un semnal de exit pentru ca serverul sa il stearga din lista clientilor activi.

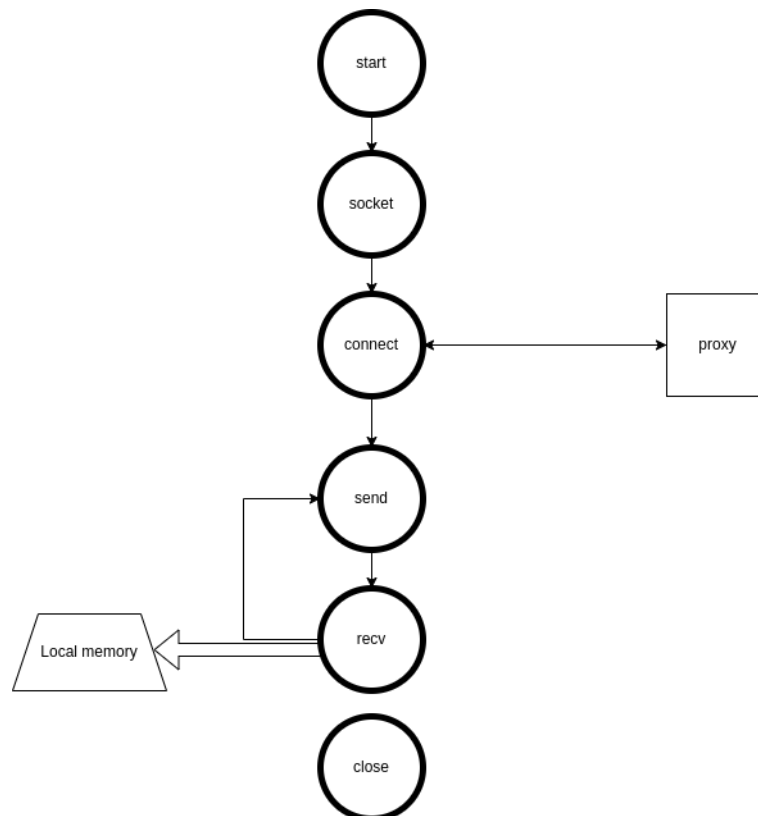


Figure 1: Schema client

Serverul proxy:

Proxy-ul asteapta conexiuni cu clientii si diverse comenzi de la acestia. De asemenea serverul se va putea conecta prin internet la un server ftp pe care il va interoga asupra operatiilor cerute de clienti(e.g. listare fisiere, preluare fisiere). La fiecare request proxy-ul trece prin fisierul de interdictii si in cazul in care nu se impune nicio restrictie comanda se initializeaza, altfel se returneaza un mesaj informativ.

Protocolul intre proxy si serverul la distanta este strict bazat pe documentatia *RFC 959*, in caz contrar ar aparea eroare pentru 'protocol mismatch'.

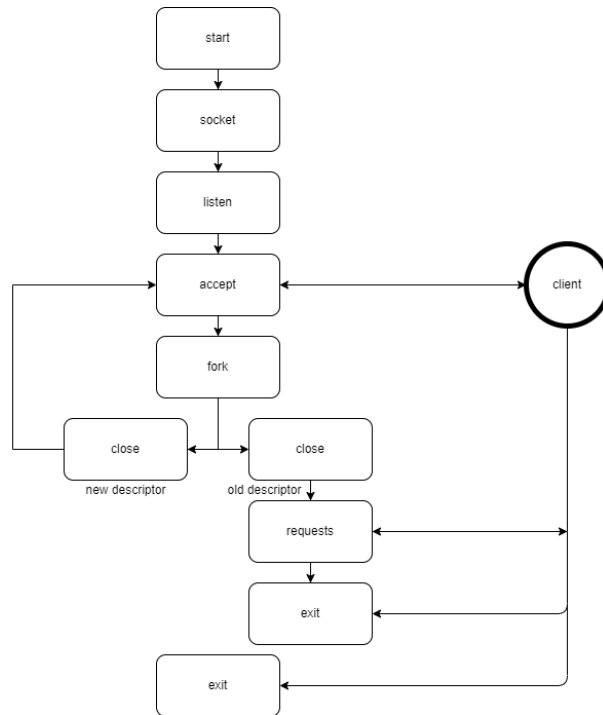


Figure 2: Schema proxy-client

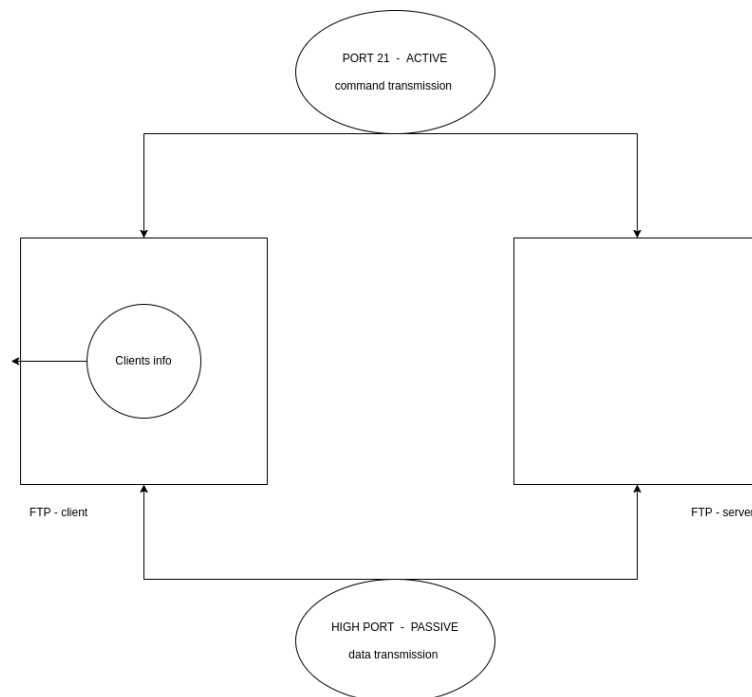


Figure 3: Schema ftp

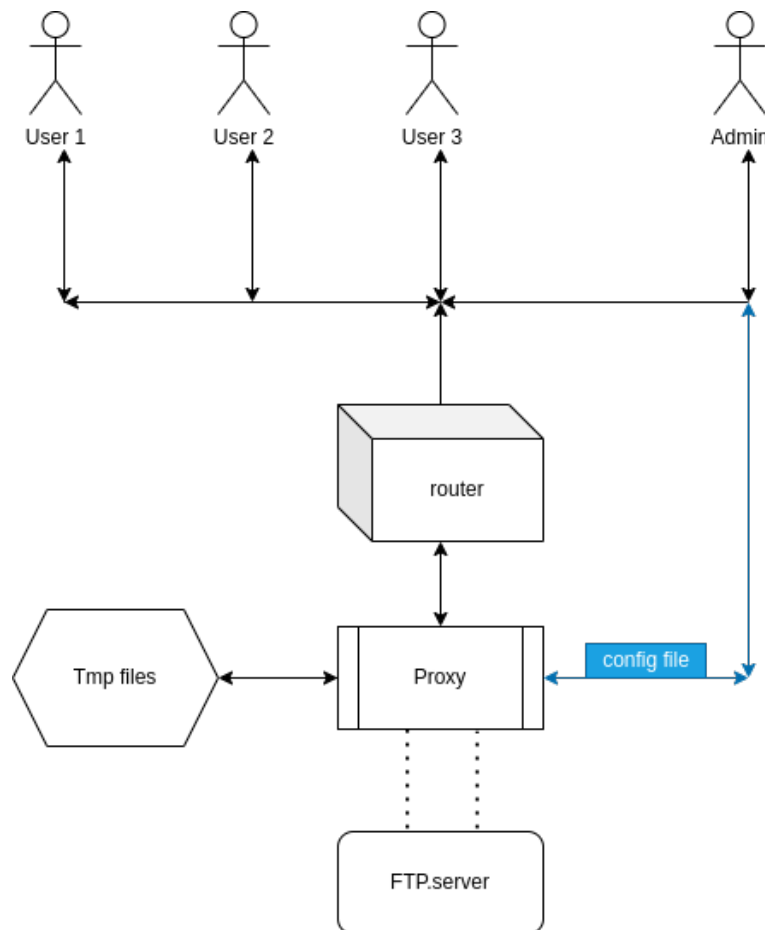


Figure 4: Schema overall

4 Detalii implementare

Atunci cand un client intra in aplicatie se va afisa un mesaj de welcome, o lista cu cateva site-uri FTP disponibile si va fi rugat sa se autentifice sau sa se inregistreze in aplicatie. In caz contrar, accesul la celelalte comenzi ii va fi restrictionat prin mesaje de eroare. Utilizatorul poate introduce orice alt site FTP dar daca protocolul serverului difera se va transmite eroarea 'protocol mismatch'.

Un client va putea astfel interoga serverul respectiv asupra fisierelor continute de acestea, dimensiunea acestora si va putea initia download-ul lor, bineinteles prin intermediul proxy-ului. In cazul in care accesul nu ii este permis, un cod de eroare corespunzator problemei ii va fi returnat. Comenzile din client trebuie sa respecte un anumit format care va fi afisat la intrarea in aplicatie, alte comenzi ducand la mesajul de eroare: 'Command unknown'.

Deoarece scopul principal este preluarea de fisiere si nu upload-ul, proxy-ul isi va face login-ul pe site-ul ftp cu username: anonymous si password: anonymous. Se poate observa in bucatile de cod cum serverul ftp intoarce raspunsuri precedate de un cod format din 3 cifre, fiecare avand o semnificatie bine determinata, ce se gaseste in *RFC 959*. Raspunsurile pe randuri multiple vor avea dupa codul din 3 cifre un caracter '-', pana la aparitia ultimei linii in care codul are un spatiu imediat urmator. In cazul intrarii in modul de preluare fisiere, serverul ii trimite proxy-ului un ip(4 numere separate prin virgula) si un port(ultimele 2 numere din sir ce se vor insuma) ce vor trebui folosite pentru a initia o conexiune socket in modul pasiv.

Fisierul json de configurare a restrictiilor va contine atat dimensiunea maxima permisa pentru download, tipul fisierelor, iar pentru fiecare server FTP cate o structura formata din: cine poate descarca de pe acesta, dar si un interval orar in care le este interzis accesul clientilor.

```

Welcome to my File Transfer client!
Here is a list with ftp site suggestions(they are rare):
    freebsd.cs.nctu.edu.tw
    ftp.pureftpd.org
    ftp.gnu.org

//////////
Connected to server.
You may introduce one command from the following list:
    login: [username]
    create: [username] [password]
    exit

You must login before using additional commands.
Please insert your command:
login: admin
Please insert password command like this:
    pass: [password]
pass: admin
User succesfully logged in.
You may introduce one command from the following list:
    logout
    server: [server-name]
    create-admin: [username] [password]
    forbidden
    exit-proxy
    exit
Please insert your command:
logout
Success. User logged out.

```

Figure 5: Interfata de comunicare client → server

```

Insert your command here: move-down: pub
Current working directory:pub

Insert your command here: list-all
The files with their lines beginning with 'd' are directories.
You can download every other file(that does not have 'd' as their first letter in line).
List of all directories and files available in the current working directory:
drwxr-xr-x  7 888      2010      14 Jan  1 18:00 FreeBSD
-rw-r--r--  1 0       2010      281 May 28 2017 README.TXT
drwxr-xr-x 1463 888    2010      177430 Oct  1 06:00 distfiles

Insert your command here: list-files
List of files and directories names:
FreeBSD
README.TXT
distfiles

Insert your command here: help
The following SITE commands are recognized
ALIAS
CHMOD
IDLE
UTIME

Insert your command here: download: README.TXT
Good to go
Download complete.
Downloaded 281/281 bytes.

```

Figure 6: Comenzi de la client la serverul FTP

```

char msg[150];
if (clients_count >= MAX_CLIENTS)
{
    strcpy(msg, "Full server right now. Try again later.");

    if (-1 == send(client, msg, strlen(msg) + 1, 0))
    {
        printf("[proxy]Error sending welcome connected to client.\n");
        exit(0);
    }

    close(client);
    continue;
}

int pid;
if ((pid = fork()) == -1)
{
    close(client);
    continue;
}
else if (pid > 0)
{
    // parinte

    for (int i = 0; i < MAX_CLIENTS; ++i)
    {
        if (clients_pid[i] == -1) // slot liber
        {
            strcpy(msg, "Connected to server.");
            if (-1 == send(client, msg, strlen(msg) + 1, 0))
            {
                printf("[proxy]Error sending welcome connected to client.\n");
                exit(0);
            }
        }
    }
}

```

Figure 7: Concurenta pe serverul proxy

```

int command_sequence(int *sd)
{
    char command[150];

    while (1)
    {
        if (USER_LOGGED == 0 && ADMIN_LOGGED == 0) // noone logged in
        {
            int command_count = 3;
            char *command_list[] = {"login: [username]",
                                    "create: [username] [password]",
                                    "exit"};

            printf("You may introduce one command from the following list:\n");
            for (int i = 0; i < command_count; ++i)
            {
                printf("\t\t%s\n", command_list[i]);
            }
            printf("\nYou must login before using additional commands.\n");

            printf("Please insert your command:\n");
            fgets(command, 150, stdin);

            mode00(command, (*sd));
        }
        else if (USER_LOGGED == 1 && ADMIN_LOGGED == 0) // normal user--
        else if (USER_LOGGED == 1 && ADMIN_LOGGED == 1) // admin user--
    }
}

```

Figure 8: Restrictionare comenzi disponibile in functie de tipul de user conectat

```

while (1)
{
    explicit_bzero(command, MAX_COMMAND);

    // citire comanda
    if (recv(client, command, MAX_COMMAND, 0) <= 0)
    {
        perror("[proxy]Eroare la read() comanda de la client.\n");
        exit(0);
    }

    printf("[proxy]Comanda a fost receptionata...%s\n", command);

    if (strncmp(command, "login: ", 7) == 0) ...
    else if (strncmp(command, "create: ", 8) == 0) ...
    else if (strncmp(command, "logout", 6) == 0) ...
    else if (strncmp(command, "server: ", 8) == 0) //aici incepe partea ftp...
    else if (strncmp(command, "create-admin: ", 14) == 0) ...
    else if (strncmp(command, "forbidden", 9) == 0) ...
    else if (strncmp(command, "exit-proxy", 10) == 0) ...
    else if (strncmp(command, "exit", 4) == 0) ...
}

```

Figure 9: Maxim de comenzi disponibile unui client

```

void my_handler(int signum)
{
    if (signum == SIGUSR1)
    {
        for (int i = 0; i < clients_count; ++i)
        {
            if (clients_pid[i] != -1)
            {
                kill(clients_pid[i], SIGTERM);
                clients_pid[i] = -1;
            }
        }
        exit(0);
    }
}

```

Figure 10: Terminare fii clienti la trimiterea exit-proxy de catre admin

```

cJSON *maxsize = NULL;
cJSON *filesallowed = NULL;
cJSON *serverdomains = NULL;

cJSON *restrictii = cJSON_Parse(file);
if(restrictii == NULL)
{
    return -1;
}

maxsize = cJSON_GetObjectItemCaseSensitive(restrictii,"maxsize");
filesallowed=cJSON_GetObjectItemCaseSensitive(restrictii,"filesallowed");

```

Figure 11: Cod prelucrare fisier JSON

```

Do you want to add other types of files?[y\n]
Please insert your answer: n
Do you want to add other domain restriction?[y\n]

Please insert your answer: y
Please insert one at a time domain structure. If you want to stop insert 'null'. Insert domain name with lowercase letters only.(e.g '.uaic.ro')
Insert here2: test.ro
Please insert one at a time client names. If you want to stop insert 'null'. Insert client name with lowercase letters only.(e.g '.y.ro')
Insert here3: testclient.ro
Please insert one at a time client names. If you want to stop insert 'null'. Insert client name with lowercase letters only.(e.g '.y.ro')
Insert here3: test2client.ro
Please insert one at a time client names. If you want to stop insert 'null'. Insert client name with lowercase letters only.(e.g '.y.ro')
Insert here3: null
Please insert a string contain digits between 1-7(1:monday,...,7:sunday). Other digits will be ignored.
Insert here4: 2346
Please insert one at a time hours. If you want to stop insert 'null'. Insert hours between 0 and 24.(e.g '17')
Insert here5: 15
Please insert one at a time hours. If you want to stop insert 'null'. Insert hours between 0 and 24.(e.g '17')
Insert here5: 10
Please insert one at a time hours. If you want to stop insert 'null'. Insert hours between 0 and 24.(e.g '17')
Insert here5: 3
Please insert one at a time hours. If you want to stop insert 'null'. Insert hours between 0 and 24.(e.g '17')
Insert here5: null
Please insert one at a time domain structure. If you want to stop insert 'null'. Insert domain name with lowercase letters only.(e.g '.uaic.ro')
Insert here2: null
Altering the configuration file completed.

```

Figure 12: Protocol modificare fisier JSON de catre admin

```

sqlite3 *db;
char *zErrMsg = 0;
int rc;
char sql[150];
char data[MAX_USERPASS];

rc = sqlite3_open(database_path, &db);

if (rc)
{
    printf("[proxy]Can't open database: %s\n", sqlite3_errmsg(db));
    exit(0);
}
else
{
    // database opened
}

sprintf(sql, "SELECT password from USERS WHERE username='%s';", utilizator);

rc = sqlite3_exec(db, sql, callback, (void *)data, &zErrMsg);

if (rc != SQLITE_OK)
{
    printf("[proxy]SQL error: %s\n", zErrMsg);
    sqlite3_free(zErrMsg);
    exit(0);
}
else
{
    // successful
}

```

Figure 13: Cod prelucrare baza de date a userilor

5 Concluzii

Aplicatia prezentata poate fi considerata un inceput pentru un program de protectie a unei retele locale impotriva datelor daunatoare ce se pot gasi pe internet si de restrictionare a accesului nedorit din partea utilizatorilor din reseaua proprie. Poate fi asemanata cu un VPN in devenire deoarece protejeaza IP-urile clientilor in reseaua de internet, dar nu la fel de sigura deoarece nu foloseste criptarea datelor in comunicarea cu exteriorul si lucreaza doar la nivel de aplicatie.

Cateva imbunatatiri care pot fi aduse proiectului ar fi legate de o interfata grafica, prietenoasa cu utilizatorii in procesul extragerii de date, dar si adaptarea la diferitele protocoale existente pe internet, un exemplu fiind https-ul si sftp-ul, deoarece protocolul ftp este unul din primele protocoale aparute, dar caruia ii lipseste un factor foarte important in zilele noastre: securitatea. Aceasta imbunatatire ar fi una cu prioritate maxima, deoarece tot mai multe browsere renunta la suportul ftp.

6 Bibliografie

Site-uri utilizate:

Protocol FTP:

1. <https://www.filestash.app/free-ftp.html>
2. <https://www.ietf.org/rfc/rfc959.txt>
3. https://enterprisedt.com/publications/FTP_Overview.html

JSON:

4. <https://github.com/DaveGamble/cJSON>

Proxy:

5. <https://nordvpn.com/blog/vpn-vs-proxy/>
6. <https://www.fortinet.com/resources/cyberglossary/proxy-server>

Baza de date:

7. <https://www.sqlite.org/cintro.html>

Comunicare server-client concurenta:

8. <https://profs.info.uaic.ro/~gcalancea/lab7/servTcpConc.c>
9. <https://profs.info.uaic.ro/~gcalancea/lab7/cliTcpConc.c>

Format LNCS:

10. <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>

Testing:

11. <https://ftp.gnu.org/>
12. <https://ftp.pureftpd.org/>
13. <https://freebsd.cs.nctu.edu.tw/>
14. <http://ftp.vim.org/>