Raport Tehnic

Ștefănică Cătălin-George

Universitatea Alexandru-Ioan Cuza, Iași Bulevardul Carol I nr. 11

1 Introducere

1.1 Proiectul ales

GridAdmin(A)
Sa se dezvolte o aplicatie care va permite administrarea unei retele de calculatoare. IP-urile statiilor din retea vor fi salvate intr-un fisier text iar aplicatia va permite executia unor comenzi pe toate pc-urile din lista. Pe toate statiile din lista se presupune ca exista instalat openssh-server si este inregistrat un administrator unic (user:admin, parola:adminpass). Lista de comenzi minime: interogarea starii pc-urilor (pornit/oprit), pornirea (wol), oprirea (shutdown), executia unor comenzi bash (cu sau fara confirmarea executiei).

1.2 Motivația

Această aplicație va putea monitoriza și controla cu ușurință calculatoarele dintr-un laborator de informatică de exemplu.Am putut observa cum funcționează o astfel de aplicație întrucât in perioada liceului, profesorul meu de informatică a utilizat un program asemănător.

1.3 Aplicabilitate

Această aplicație va putea fi folosită de oricine dorește să monitorizeze o rețea de calculatoare, fie aceasta un laborator de informatică sau chiar o rețea locală dintr-o companie.

2 Tehnologii Utilizate

2.1 TCP

TCP este un protocol de transport orientat conexiune,fara pierdere de informații ce vizează oferirea calității maxime a serviciilor. Acesta utilizează conexiuni , nu porturi ca abstracțiuni fundamentale. Ambele părți (expeditorul și destinatarul) trebuie să participe la realizarea conexiunii ,aceasta fiind identificată prin perechi reprezentate de adresă IP:PORT(socket) .

2.2 Secure Shell [SSH]

Secure Shell (SSH) este un protocol de rețea criptografic ce permite ca datele să fie transferate folosind un canal securizat intre dispozitive de rețea. Cele două mari versiuni ale protocolului sunt SSH1 sau SSH-1 și SSH2 sau SSH-2.

Folosit cu precădere în sistemele de operare multiutilizator linux și unix, SSH a fost dezvoltat ca un înlocuitor al Telnet-ului și al altor protocoale nesigure de acces de la distanță, care trimit informatia, în special parola, în clar, făcând posibilă descoperirea ei prin analiza traficului.

Criptarea folosită de SSH intenționează să asigure confidențialitatea și integritatea datelor transmise printr-o rețea nesigură cum este Internetul.

3 Arhitectura Aplicației

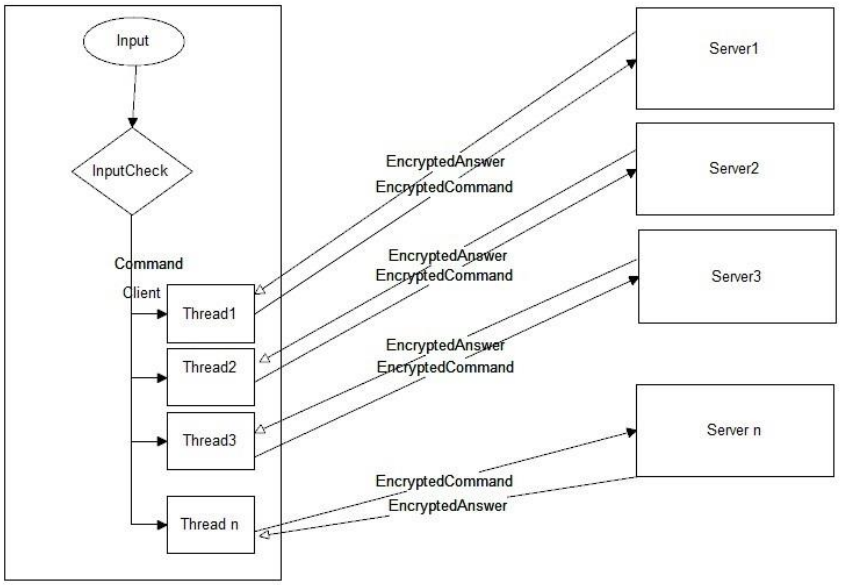
3.1 Conceptele implicate

Initial trebuie salvate adresele IP ale calculatoarelor din reteaua pe care dorim să o administrăm. Acestea vor fi salvate intr-un fisier text pe calculatorul administratorului.

Clientul va inițializa câte o conexiune cu fiecare calculator din listă in mod concurrent. Se va utiliza cate un thread pentru fiecare conexiune efectuată. Conexiunea si comunicarea va fi realizată prin intermediul SSH, fiecare calculator din listă având openssh-server instalat si un administrator unic (user:admin și pass:adminpass).

Clientul va introduce date de la tastatură iar acestea vor fi trimise la server(e). Astfel clientul va putea trimite comenzi la intreaga rețea sau la un singur calculator . Datele transmise for fi criptate inainte de transmisie de către client si decriptate când ajung la server.

3.2 Diagrama



4 Detalii de implementare.

4.1 Cod Asociat Proiectului

```c
#include <libssh/libssh.h>
#include <stdlib.h>

int main()
{
  ssh_session my_ssh_session;
  int verbosity = SSH_LOG_PROTOCOL;
  int port = 22;

  my_ssh_session = ssh_new();
  if (my_ssh_session == NULL)
    exit(-1);

  ssh_options_set(my_ssh_session, SSH_OPTIONS_HOST, "localhost");
  ssh_options_set(my_ssh_session, SSH_OPTIONS_LOG_VERBOSITY, &verbosity);
  ssh_options_set(my_ssh_session, SSH_OPTIONS_PORT, &port);
```

Fiecare thread va conține codul pentru realizarea unei conexiuni SSH. In cazul de mai sus “localhost” va fi înlocuit cu adresa IP a unui calculator de retea si de asemnea va fi adaugata optiunea: **ssh_options_set(my_ssh_session, SSH_OPTIONS_USER , “admin”)** întrucât va trebui să ne conectăm drept administrator.

```
switch (state) {
    case SSH_KNOWN_HOSTS_OK:
        /* OK */

        break;
    case SSH_KNOWN_HOSTS_CHANGED:
        fprintf(stderr, "Host key for server changed: it is now:\n");
        ssh_print_hexa("Public key hash", hash, hlen);
        fprintf(stderr, "For security reasons, connection will be stopped\n");
        ssh_clean_pubkey_hash(&hash);

        return -1;
    case SSH_KNOWN_HOSTS_OTHER:
        fprintf(stderr, "The host key for this server was not found but an other"
                "type of key exists.\n");
        fprintf(stderr, "An attacker might change the default server key to"
                "confuse your client into thinking the key does not exist\n");
        ssh_clean_pubkey_hash(&hash);

        return -1;
    case SSH_KNOWN_HOSTS_NOT_FOUND:
        fprintf(stderr, "Could not find known host file.\n");
        fprintf(stderr, "If you accept the host key here, the file will be"
                "automatically created.\n");

        /* FALL THROUGH to SSH_SERVER_NOT_KNOWN behavior */

    case SSH_KNOWN_HOSTS_UNKNOWN:
        hexa = ssh_get_hexa(hash, hlen);
        fprintf(stderr,"The server is unknown. Do you trust the host key?\n");
        fprintf(stderr, "Public key hash: %s\n", hexa);
        ssh_string_free_char(hexa);
        ssh_clean_pubkey_hash(&hash);
        p = fgets(buf, sizeof(buf), stdin);
        if (p == NULL) {
            return -1;
        }
```

Foarte important pentru aplicatie va fi verificarea server-ului inainte de conexiune pentru a ne asigura ca acesta este unul cunoscut si sigur.

4.2 Scenarii de utilizare

Clientul se va conecta la serverele din listă si va introduce în mod automat user-ul și parola specifică. In cazul în care un server nu este active sau a întâmpinat o problem la conexiune aceasta va fi afișată și va fi menționat IP-ul serverului problematic.

În momentul în care s-a conectat cu success la servere, acestea vor astepta să primească comenzi de la client. Dacă clientul va incerca o comandă invalid va apărea un mesaj sugestiv.Dacă comandă este validă aceasta este transmisă criptată la toate serverele sau la cel/cele specificate, după cum a dorit clientul.

Serverul va decripta comanda primită si va verifica de asemenea ca aceasta sa fie validă. Va executa comanda dacă nu a găsit vreo problemă.

5 Concluzie

Întrucât se utilizeaza SSH, aplicația va oferi utilizatorului un mediu foarte sigur, ascultarea traficului sau interferența din afara fiind aproape imposibilă datorită criptării datelor transmise si necesității de a folosi parole la conectare.

Pentru îmbunătățirea proiectului, se poate creea o functie în aplicație care verifică la un anumit interval de timp dacă unul dintre calculatoarele din rețea ,initial inactive, a fost pornit si va inițializa în mod automat conectarea la acesta.

Pentru îmbunătățirea securității se poate utiliza autentificarea prin chei publice/private si schimbate astfel parolele de la administrator pentru a fi unice si greu de descoperit.

References

https://profs.info.uaic.ro/~computernetworks/

https://api.libssh.org/stable/libssh_tutor_guided_tour.html