**Remote Administration Tool (RAT) - Documentație de Proiect**

**Autori**: Pila Alexandru & Ivescu Mircea

**Curs**: Proiectarea Sistemelor de Operare

**Cuprins**

1. **Introducere**
   * Scopul Documentului
   * Prezentare Generală a Aplicației
2. **Descrierea Sistemului**
   * Arhitectura Generală
   * Tehnologii și Limbaje Utilizate
3. **Funcționalități și Caracteristici**
   * Funcționalități de Bază
     + Scanner de Parole în Fișiere
     + Execuție Remote
     + Preluare Informații Sistem
     + Screenshot

**1. Introducere**

**Scopul Documentului**

Acest document detaliază procesul de dezvoltare a unei aplicații de administrare remote (RAT) pe platforma Linux, utilizând limbajul C.

**Prezentare Generală a Aplicației**

Aplicația noastră este un instrument de administrare remote conceput pentru a permite controlul complet al unui sistem Linux de la distanță. Aceasta va oferi funcționalități de bază, precum execuția de comenzi și transferul de fișiere, dar și caracteristici avansate( scanner de parole, monitorizare trafic, etc.)

**2. Descrierea Sistemului**

**Arhitectura Generală**

Aplicația se bazează pe o arhitectură client-server:

* **Client**: Aplicatie aflata pe o statie, care raspunde comenzilor de pe server
* **Server**: Gestionarea mai multor clienti, fiecare pe un thread propriu, cu posibilitatea de a transmite comenzi si de a monitoriza date utile de pe fiecare client ( o eventuala interfata grafica )
* Client și Server comunică prin socket-uri TCP.
* Comunicația este criptată pentru a asigura securitatea(???).

**Tehnologii și Limbaje Utilizate**

* **Limbaj de Programare**: C
* **Platformă**: Linux
* **Biblioteci**:
  + **POSIX Threads** pentru gestionarea multi-threading.
  + **Sockets API** pentru comunicarea între client și server.

**3. Funcționalități și Caracteristici**

**Funcționalități de Bază**

**Monitorizarea clientilor**

* **Descriere:** Permite serverului sa vada toti clientii conectati si ofera posibilitatea de gestionare
* **Implementare:** Fiecare conexiune client este gestionată printr-un thread sau un proces separat, asigurând astfel că serverul poate urmări activitatea fiecărui client în timp real. Implementarea unei interfețe în linia de comandă care permite vizualizarea clienților activi și accesul la acțiuni de gestionare (deconectare, blocare, etc.)
* **Abordare:** La fiecare conectare a unui client, serverul adaugă o nouă intrare în structura de date a clienților conectați. Se implementează un mecanism de monitorizare continuă care actualizează în timp real starea fiecărui client. De exemplu, un thread dedicat poate verifica activitatea clienților și poate trimite ping-uri de verificare.

**Execuție de Comenzi Remote**

* **Descriere**: Permite clientului să ruleze comenzi shell pe sistemul remote și să primească rezultatele.
* **Implementare**: Folosirea funcțiilor system() sau execvp() pentru a rula comenzile.
* **Abordare**:
  + Clientul trimite comanda ca un string prin socket.
  + Serverul primește comanda, o execută folosind system() și returnează rezultatul către client.

**Transfer de Fișiere**

* **Descriere**: Permite clientului să încarce sau să descarce fișiere de pe server.
* **Implementare**: Folosirea funcțiilor read() și write() pentru transferul de fișiere.
* **Abordare**:
  + Clientul specifică fișierul pe care îl dorește.
  + Serverul primește solicitarea și transferă fișierul.

**Monitorizare Sistem**

* **Descriere**: Clientul poate vizualiza utilizarea CPU, memorie și alte statistici de sistem.
* **Implementare**: Utilizarea funcțiilor standard din Linux pentru preluarea datelor din /proc.
* **Abordare**:
  + Serverul citește fișierele relevante din /proc și trimite informațiile clientului.

**Scanner de Parole în Fișiere**

* **Descriere**: Scanează fișierele pentru a identifica credențialele userilor din sistem (din /etc/passwd și /etc/shadow)
* **Implementare**: Folosirea funcțiilor de citire a fișierelor și regex pentru detectarea userilor și hash-ul parolelor acestora.

**Preluare Informații Sistem**

* **Descriere**: Colectează date despre sistemul de operare, hardware, utilizatori și procese active.
* **Implementare**: Comenzi Linux (ex. uname, who, ps) combinate cu funcțiile C pentru parsing.
* **Abordare**:
  + Serverul rulează comenzile specifice și colectează informațiile.
  + Datele sunt trimise într-un format structurat către client pentru afișare.

**Screenshot**

* **Descriere**: Captura de ecran a unui client
* **Implementare**: Folosirea unui utilitar și transmiterea imaginii ca un fișier
* **Abordare**:
  + Serverul cere un screenshot iar clientul îl face și îl trimite.
* **Monitorizare Rețea în Timp Real**: Afișarea conexiunilor active și a traficului.