APLICAȚIE

TIP PORT SCANNER

****

Îndrumător: Realizat de:

Slt. Vaman Adina Sd. Sg. Călinescu Marina-Corina

Sd. Sg. Modiga Miruna-Mihaela

Grupa **C113C**

**CUPRINS**

Capitol 1 - Introducere 3

* 1. [Scopul proiectului 3](#_TOC_250004)
  2. [Lista definițiilor 3](#_TOC_250003)
  3. [Descrierea generală a aplicației 4](#_TOC_250002)

[Capitolul 2 – Descrierea mediului de operare 5](#_bookmark0)

* 1. [Componente Back-end și Front-end 5](#_bookmark1)
  2. [Mediul de dezvoltare 5](#_TOC_250001)
  3. [Tabel de versiuni 5](#_bookmark1)

**Capitolul 3** – **Descrierea modelului logic al sistemului** 7

* 1. **Arhitectura aplicației** 7
  2. **Descriere funcționalități** 8

## Capitolul 4 – Testare funcționalități 9

**Capitolul 1 – Introducere**

## Scopul proiectului

Crearea unui produs software care să ofere utilizatorilor posibilitatea de a scana și monitoriza starea porturilor de rețea, permițându-le să vizualizeze traficul și să afle dacă acestea sunt deschise sau închise.

## Lista definițiilor

Un port scanner este o aplicație software prin care putem explora rețeaua sau sistemul de calculatoare pentru a determina traficul care are loc, implicit disponibilitatea porturilor.

La nivel de software, în cadrul unui sistem de operare, un port este o construcție logică care identifică un proces specific sau un tip de serviciu de rețea. Un port este identificat pentru fiecare protocol de transport și combinație de adrese printr-un număr pe 16 biți, cunoscut sub numele de număr de port.

Protocoalele folosite pentru scanarea porturilor sunt protocoalele UDP și TCP.

## Descrierea generală a aplicației

Aplicația este o unealtă de securitate și de gestionare a rețelei, dezvoltată în C, care permite utilizatorului să scaneze porturile pe o adresă IP specificată (hostname) pentru a determina starea acestora (deschise sau închise). Această aplicație oferă o funcționalitate similară comenzii **nmap** din Linux și **netstat** din Windows, permițând utilizatorilor să efectueze scanări de porturi pentru a investiga disponibilitatea și securitatea serviciilor de rețea.

Utilizatorii pot avea opțiuni suplimentare pentru a personaliza scanările, cum ar fi specificarea unui anumit port, a unui range de porturi, sau opțiunea implicită a aplicației de a scana toate porturile

# Capitolul 2 – Descrierea mediului de operare

## Componente Back-end și Front-end

Interfața grafică va fi dezvoltată folosind framework-ul Qt, iar aplicația software va fi creată folosind limbajul de programare C.

## Mediul de dezvoltare

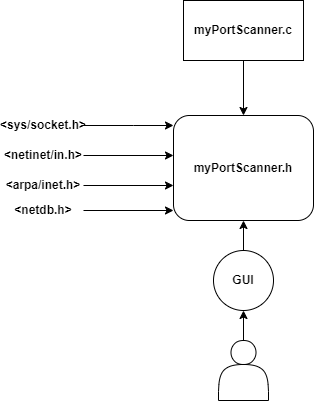
Produsul software va fi dezvoltat pentru dispozitive pe care ruleaza sistemul de operare Linux, împreună cu sistemul inter-platformă de dezvoltare Qt pentru a crea interfața grafică oferită utilizatorului.

## Tabel de versiuni

|  |  |
| --- | --- |
| **Numărul versiunii** | **Modificări adăugate** |
| **Versiunea 1** | Versiune inițială - TCP conection port scanning |
| **Versiunea 2** | TCP SYN scan - half-open scanning |
| **Versiunea 3** | Implementare cu threaduri + opțiuni suplimentare |

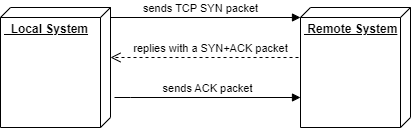
# Capitolul 3 – Descrierea modelului logic al sistemului

## 3.1. Arhitectura aplicației

****

Aplicația scanează porturile unei adrese IP, sau a unui hostname (ce este translatat folosind funcția gethostbyname() din header-ul <**netdb.h**>). Aceasta dispune de trei moduri de funcționare: scanarea unui port specific, scanarea unui range de porturi sau scanarea tuturor porturilor de rețea. Funcționalitatea este dată prin crearea de conexiuni către porturi de tip socket, utilizând header-ul <**sys/socket.h**> (TCP connect).

Scanarea porturilor utilizând metoda de conectare TCP implică stabilirea unei conexiuni complete, care presupune un schimb de trei etape între cele două gazde implicate. În cazul SYN , scanner-ul nu mai realizează ACK packet, conxiunea nefiind finalizată.

**

Rezultate scanare:

* **Open** - rețeaua țintă acceptă conexiuni sau pachete și a răspuns cu un pachet care arată că este în ascultare
* **Closed -** rețeaua a primit cererea, dar nu rulează niciun serviciu pe acel port
* **Filtered** - un pachet de cerere a fost trimis, dar gazda nu a răspuns și nu este în ascultare

## 3.2. Descriere funcționalități

După rularea programului, se va deschide o interfață grafică, prietenoasă cu utilizatorul. Aplicația este reprezentată printr-o fereastră principală cu diferite căsuțe ce trebuiesc completate de utilizator pentru a stabili opțiunile scanării, precum si un buton ce pornește scanarea porturilor specificate.

**Fereastra principală**:

Implicit, aplicația va scana toate porturile de la 1 la 1024.

TextBox “Ports”: scanarea se va efectua pe un port/range de porturi.

TextBox “Target”: permite utilizatorului să scaneze o adresă IP individuală sau un hostname ce va fi tradus într-o adresa IP.

Checkboxes “Options”: utilizatorul va putea selecta opțiunile scanării

Buton “Start Scan”: permite utilizatorului să inceapa scanarea porturilor specificate

**Fișierul de output**:

După scanare, aplicația va scrie într-un fișier rezultatele scanării.

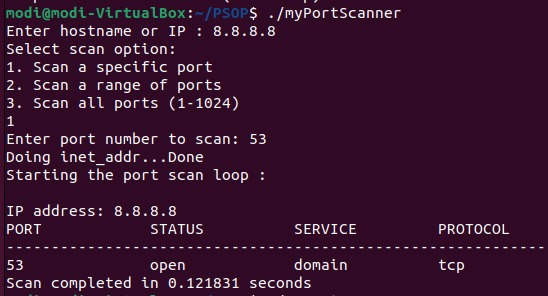
**Capitolul 4 – Testare funcționalități**

1. **Test 1** - verificare hostname valid, port specificat

INPUT:

IP Address: 8.8.8.8

Port: 53 si 443





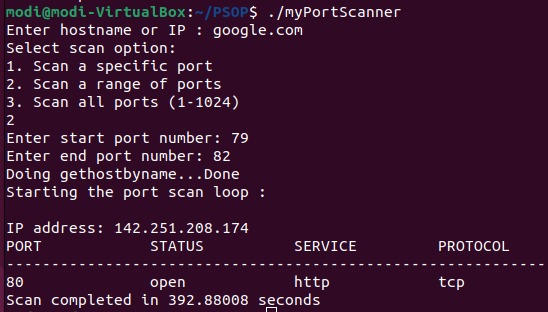
1. **Test 2** - verificare conexiune http - range de porturi

INPUT:

Hostname: google.com

Start port number: 79

End port number: 82

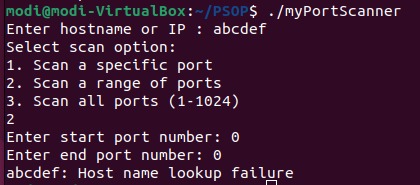


1. **Test 3** - verificare conexiune http - hostname invalid

INPUT:

Hostname: abcdef

Port: 80



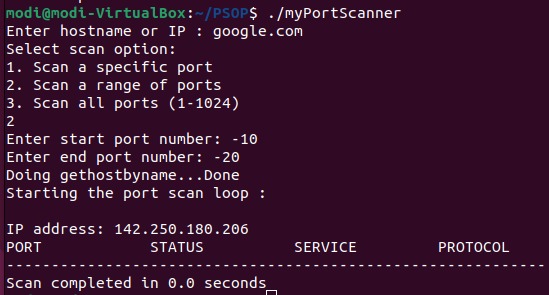
1. **Test 4** - verificare conexiune http - range de porturi invalide

INPUT:

Hostname: google.com

Start port number: -10

End port number: -20



1. **Test 5** - niciun target introdus

