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 3](#_TOC_250002)

[Capitolul 2 – Descrierea mediului de operare 4](#_bookmark0)

* 1. [Componente Back-end și Front-end 4](#_bookmark1)
  2. [Tabel de versiuni 4](#_bookmark1)

**Capitolul 3** – **Descrierea modelului logic al sistemului** 5

* 1. **Arhitectura aplicației** 5
  2. **Descriere funcționalități** 9

## Capitolul 4 – Testare funcționalități 10

**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

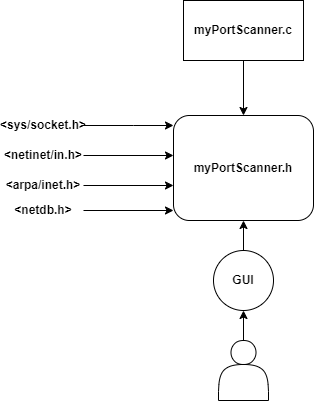
Interfața grafică va fi realizata in linia de comanda , iar aplicația software va fi creată folosind limbajul de programare C.

## Tabel de versiuni

|  |  |
| --- | --- |
| **Numărul versiunii** | **Modificări adăugate** |
| **Versiunea 1** | Versiune inițială - TCP conection port scanning |
| **Versiunea 2** | Implementare cu threaduri + opțiuni suplimentare |
| **Versiunea 3** | Tipuri de scanare: TCP, UDP |
| **Versiunea 3.2** | Tipuri de scanare: SYN, FIN, XMAS, NULL |
| **Versiunea 3.3** | Tipuri de scanare: ACK, WINDOW, CUSTOM |

# 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**>). Proiectul implementează un scanner de securitate care permite utilizatorului să efectueze scanări de porturi pe un anumit host. Acesta suportă multiple moduri de scanare, inclusiv scanarea TCP connect, scanarea SYN, scanarea UDP, scanarea NULL, scanarea FIN, scanarea XMAS, scanarea ACK, scanarea Window și scanarea personalizată.

Principalele componente și funcționalități ale aplicației includ:

* Structura de Date **”thread\_options”**:

Stochează opțiunile pentru fiecare fir de execuție, cum ar fi adresa IP a hostului, portul, timeout-ul, numărul de fire, opțiunile de scanare (**TCP**, **UDP**, **SYN**, **NULL, FIN, XMAS, ACK, WINDOW, CUSTOM**).

* Funcția **”print\_banner”**:

Afisează un banner la începutul rulării programului, furnizând o prezentare estetică.

* Funcțiile **”TCP\_scan”, ”UDP\_scan”, ”CustomScan”,**

**”SYN\_NULL\_FIN\_XMAS\_scan”:**

Implementează metodele specifice pentru fiecare tip de scanare (TCP, UDP, CUSTOM, SYN, NULL, FIN, XMAS, ACK, WINDOW).

Se realizează conexiuni sau trimit pachete specifice și se analizează răspunsurile.

* Funcția **”csum”:**

Calculează suma de control pentru verificarea integrității pachetelor.

* Funcția **”parse\_args”**:

Analizează și interpretează argumentele de la linia de comandă, stabilind opțiunile specifice ale utilizatorului.

Functia **parse\_args** foloseste biblioteca **arpg** care oferă un mecanism ușor de utilizat pentru analiza și gestionarea argumentelor liniei de comandă, permițând definirea opțiunilor și a funcțiilor asociate pentru manipularea acestora într-un mod modular și eficient.

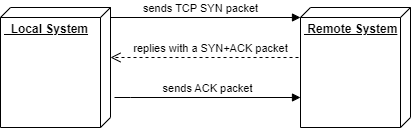
Funcția **parse\_opt** procesează opțiunile din linia de comandă, manipulând structura de argumente în funcție de codul opțiunii (**key**). Se realizează acțiuni precum setarea host-ului, timeout-ului, scanarea specifică a porturilor, și afișarea ajutorului.

* Funcțiile de scanare (TCP, UDP, SYN, NULL, FIN, XMAS, ACK, WINDOW, CUSTOM):

Folosesc **raw sockets** pentru a manipula direct pachetele la nivel de rețea. Construiesc și trimit pachete specifice, apoi așteaptă și interpretează răspunsurile.

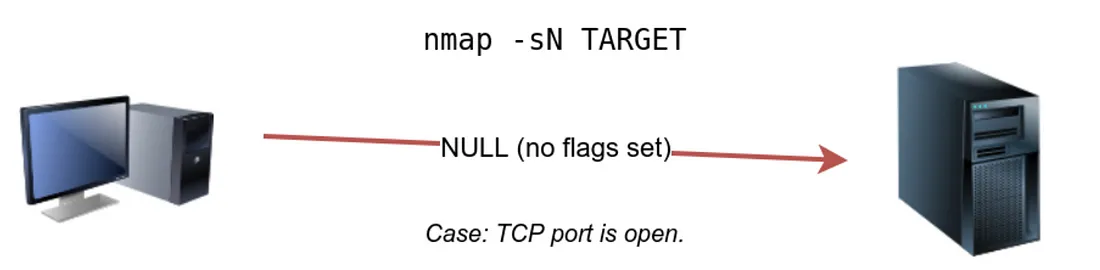
* **TCP SYN scan**:

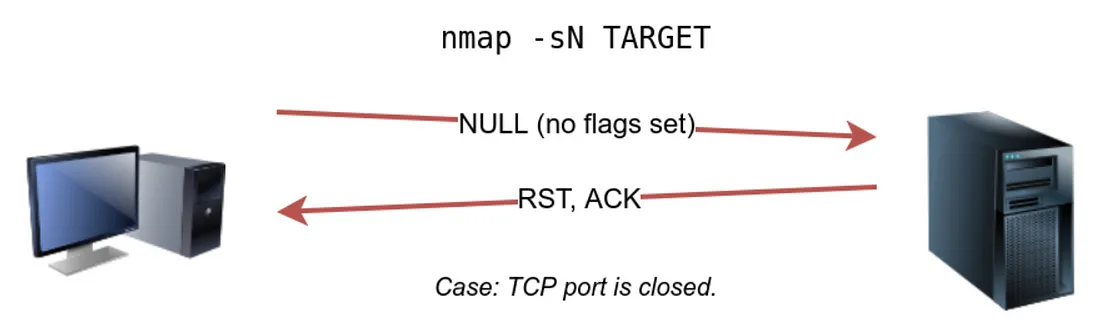
Scanarea porturilor utilizând metoda de conectare TCP implică stabilirea unei conexiuni complete, care presupune un schimb de trei etape între cele două gazde implicate. În cazul SYN , scanner-ul nu mai realizează ACK packet, conxiunea nefiind finalizată.

**

Rezultate scanare:

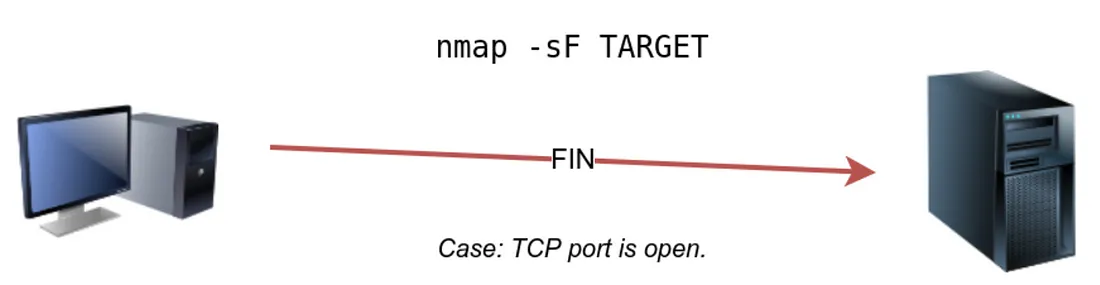
* **Open** - rețeaua țintă acceptă conexiuni sau pachete și a răspuns cu un pachet care arată că este în ascultare
* **Closed -** rețeaua a primit cererea, dar nu rulează niciun serviciu pe acel port
* **Filtered** - un pachet de cerere a fost trimis, dar gazda nu a răspuns și nu este în ascultare

****

* **NULL scan:**

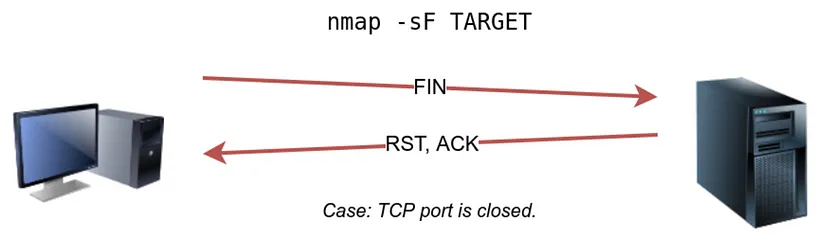
Nu seteaza niciun flag.

Toate cele 6 flag-uri sunt setate la 0.

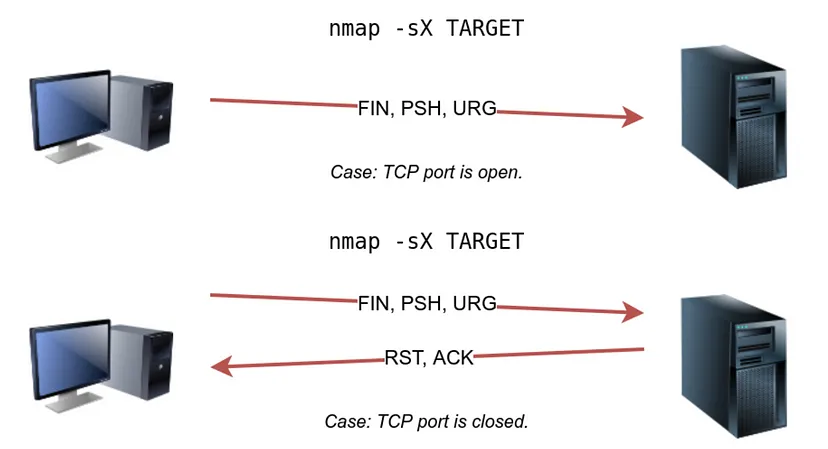


* **FIN scan:**

Transmite un pachet TCP cu flag-ul „FIN”.

**** Cu toate acestea, niciun răspuns nu va fi

trimis înapoi dacă portul TCP este deschis.



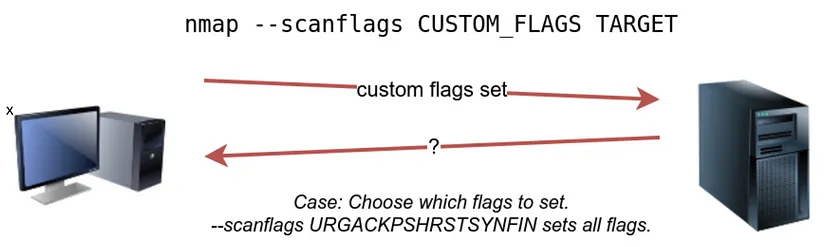
* **XMAS scan:**

Scopul principal al acestei scanări este de

a activa simultan flag-urile

FIN, PSH și URG.

|  |  |
| --- | --- |
| **Probe Response** | **Assigned State** |
| No response received (even after retransmissions) | open|filtered |
| TCP RST packet | closed |
| ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) | filtered |

****

* **CustomScan:**

## 3.2. Descriere funcționalități

După rularea programului, se va deschide o interfață grafică, prietenoasă cu utilizatorul. Aceasta este reprezentată de logo-ul aplicației in continuarea căruia, in funcție de scanarea și opțiunile alese, vor apărea tipul de scanare ales, adresa ip a cărei porturi le scanezi si porturile impreuna cu starea lor din momentul respectiv. La final este precizată și durata in care are loc scanarea.

**Scanările disponibile**:

* **TCP SYN Scan:** -sS;
* **TCP CONNECT Scan:** -sT;
* **UDP Scan:** -sU;
* **TCP FIN Scan:** -sF;
* **TCP NULL Scan:** -sN;
* **TCP XMAS Scan:** -sX;
* **TCP ACK Scan:** -sA;
* **TCP WINDOW Scan:** -sW;
* **CUSTOM Scan Types with flags:** -sC.

**Opțiunile disponibile:**

* **-h** - adresa ip/hostname;
* **-p** - portul/range-ul de porturi pe care dorim sa le scanăm;
* **-T** - numărul de thread-uri pe care dorim sa le folosim in scanare;
* **-v** - verifică tipul de serviciu găsit pe port și tipul de protocol.

**Exemplu:**

./numeleAplicatiei -h google.com -p 80 -T 1 -sS

**Capitolul 4 – Testare funcționalități**

Testele vor fi verificate cu **nmap** pentru validare.

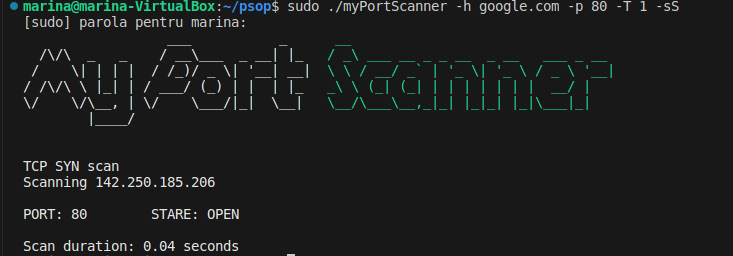
1. **Test 1** – Scanare TCP SYN Scan pe un singur port:

Input:

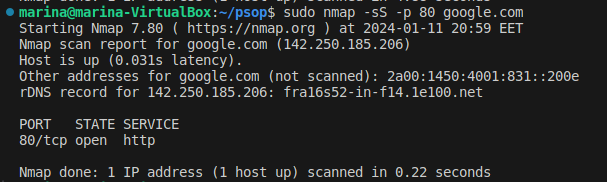
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**



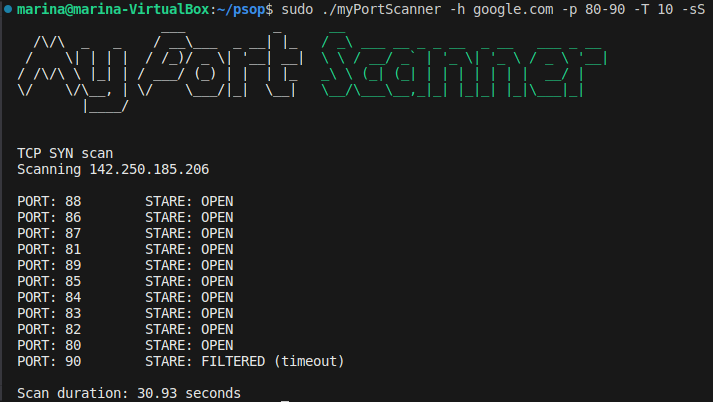
1. **Test 2** – Scanare TCP SYN Scan pe un range de porturi:

Input:

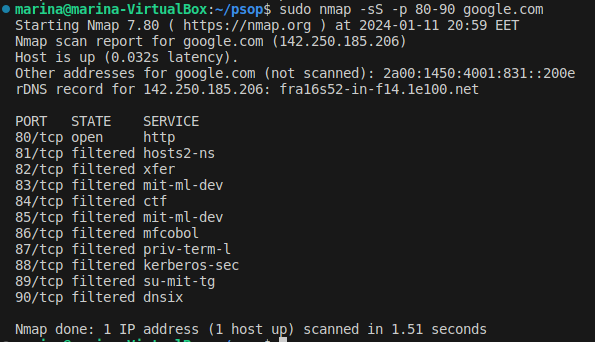
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**



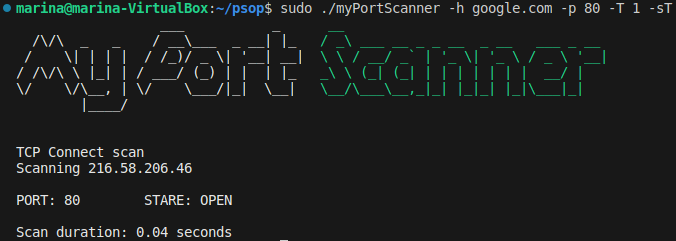
1. **Test 3** – Scanare TCP CONNECT Scan pe un singur port:

Input:

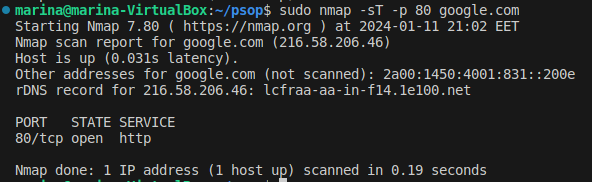
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

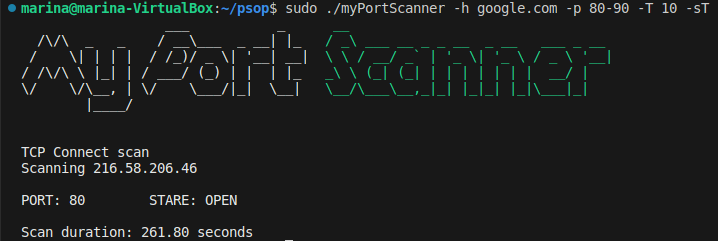
1. **Test 4** – Scanare TCP CONNECT Scan pe un range de porturi:

Input:

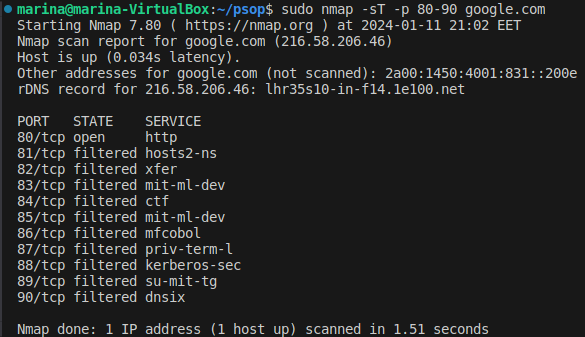
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

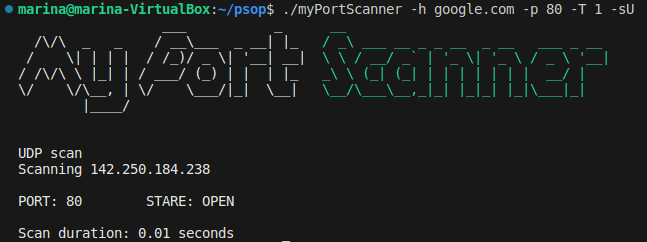
1. **Test 5** – Scanare UDP Scan pe un singur port:

Input:

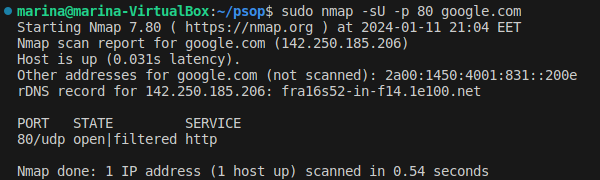
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

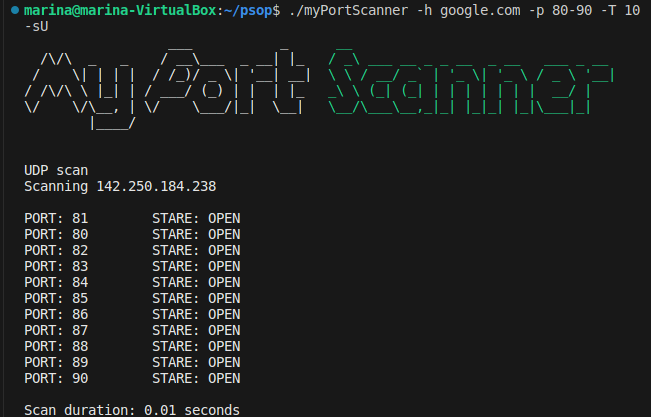
1. **Test 6** – Scanare UDP Scan pe un range de porturi:

Input:

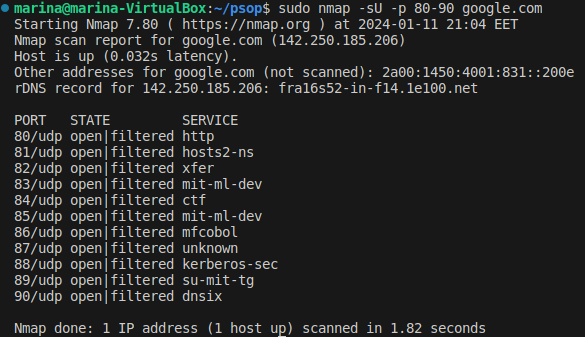
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

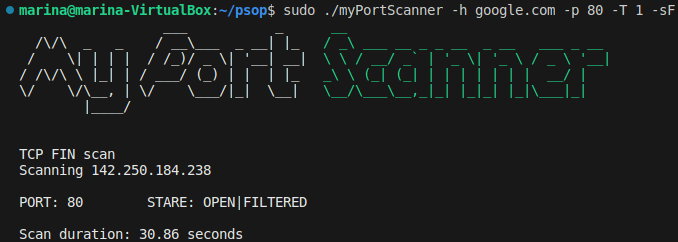
1. **Test 7** – Scanare TCP FIN Scan pe un singur port:

Input:

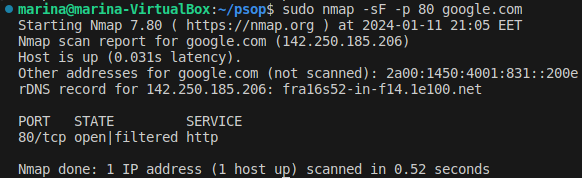
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

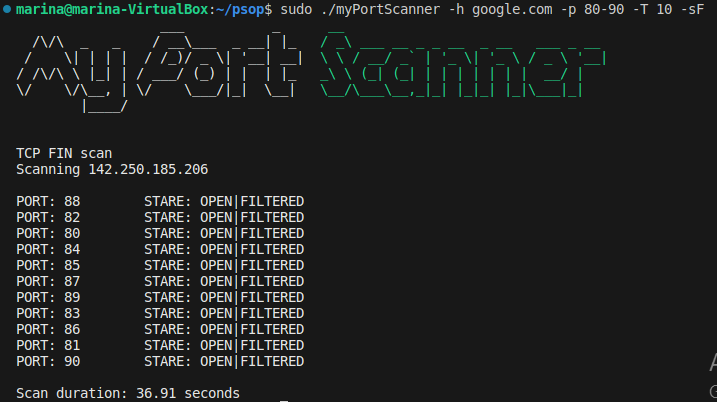
1. **Test 8** – Scanare TCP FIN Scan pe un range de porturi:

Input:

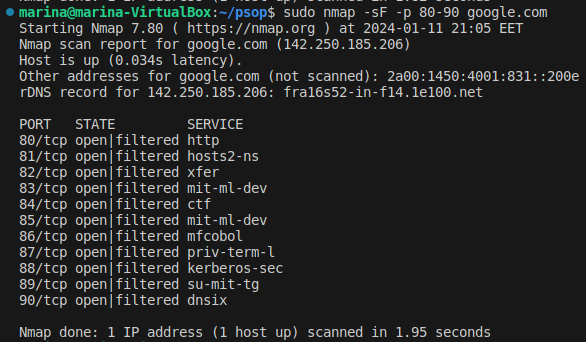
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

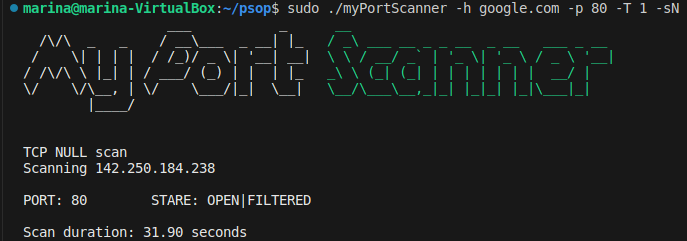
1. **Test 9** – Scanare TCP NULL Scan pe un singur port:

Input:

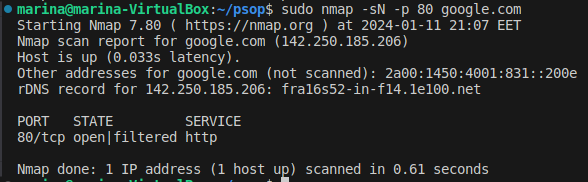
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**

****

**Nmap:**

****

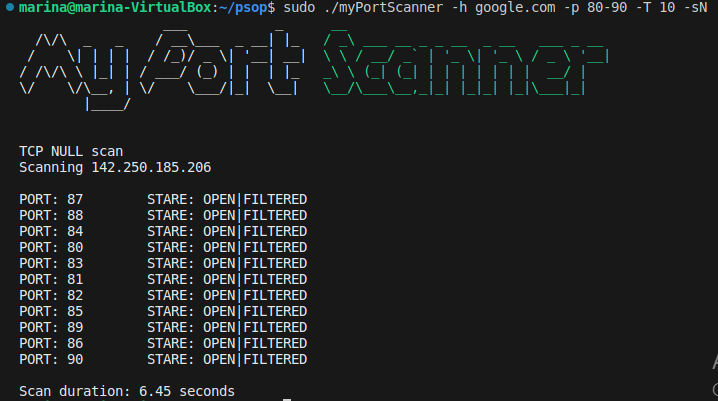
1. **Test 10** – Scanare TCP NULL Scan pe un range de porturi:

Input:

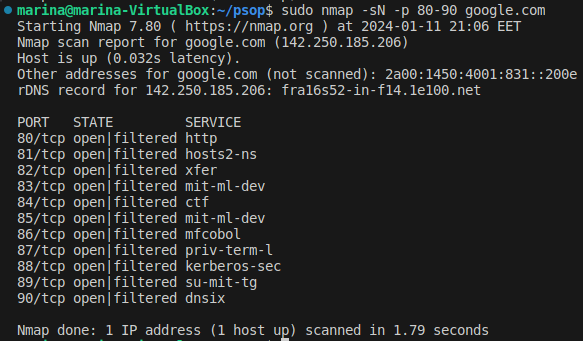
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**



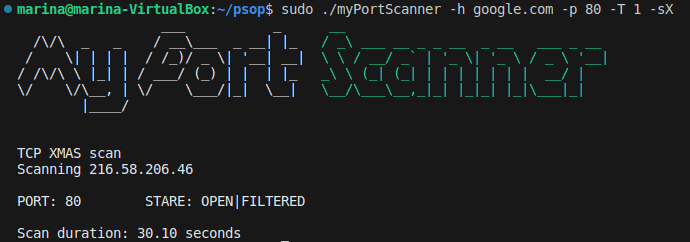
1. **Test 11** – Scanare TCP XMAS Scan pe un singur port:

Input:

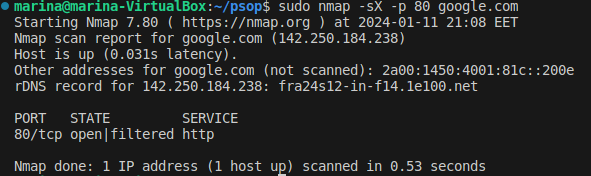
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

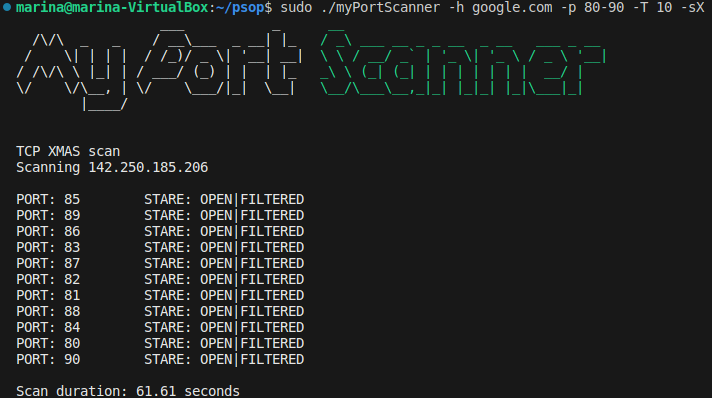
1. **Test 12** – Scanare TCP XMAS Scan pe un range de porturi:

Input:

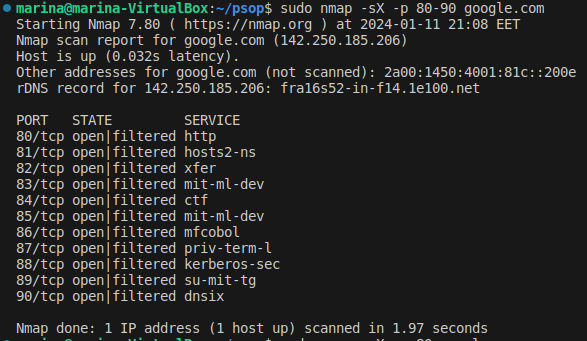
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

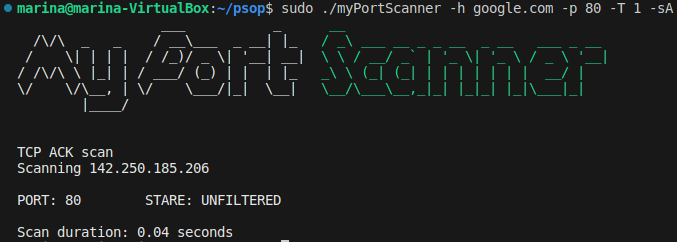
1. **Test 13** – Scanare TCP ACK Scan pe un singur port:

Input:

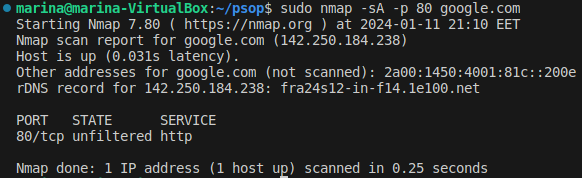
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

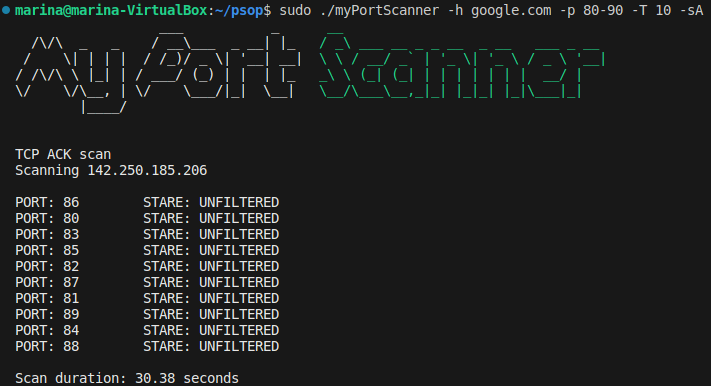
1. **Test 14** – Scanare TCP ACK Scan pe un range de porturi:

Input:

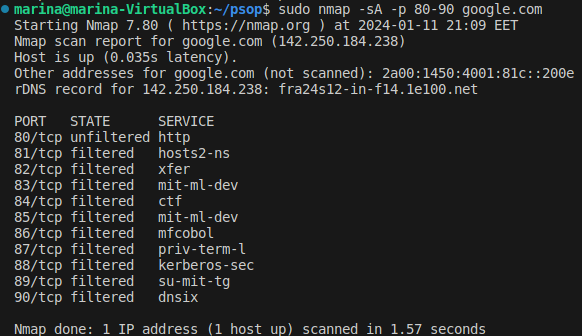
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

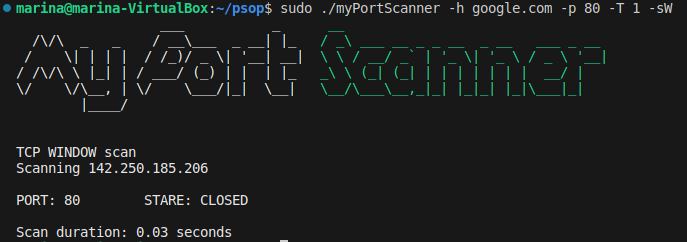
1. **Test 15** – Scanare TCP WINDOW Scan pe un singur port:

Input:

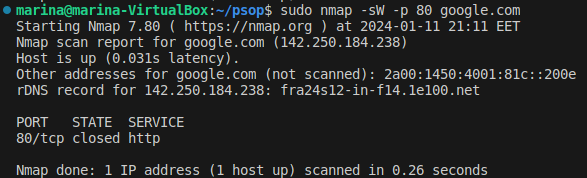
* Hostname: google.com
* Port: 80

Output:

**./myPortScanner:**



**Nmap:**

****

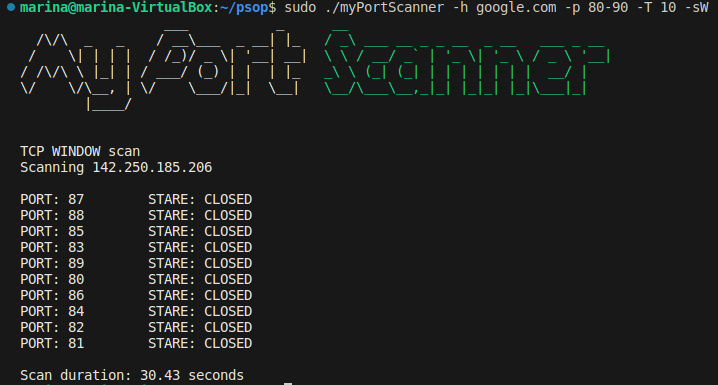
1. **Test 16** – Scanare TCP WINDOW Scan pe un range de porturi:

Input:

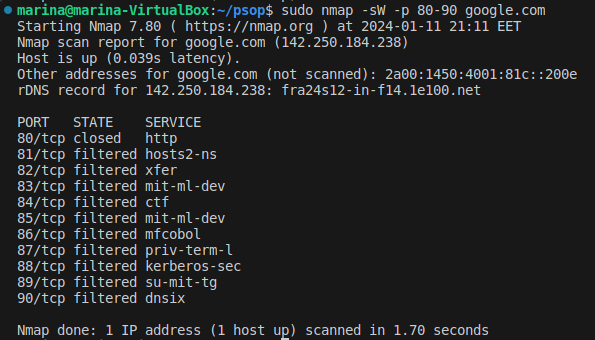
* Hostname: google.com
* Port: 80-90

Output:

**./myPortScanner:**



**Nmap:**

****

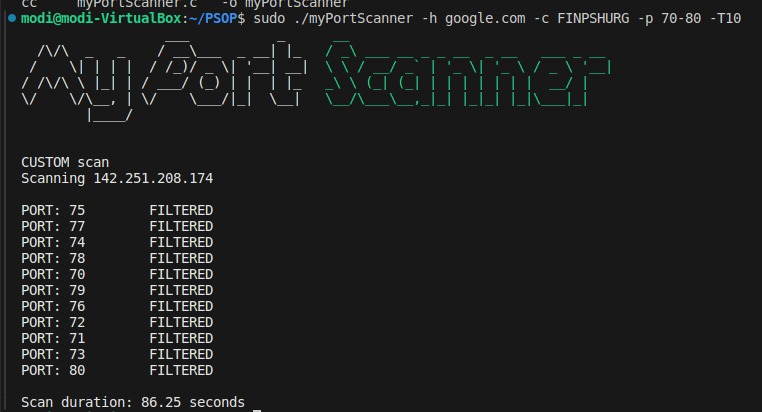
1. **Test 17** – Scanare CUSTOM pe un range de porturi(comparativ cu XMAS):

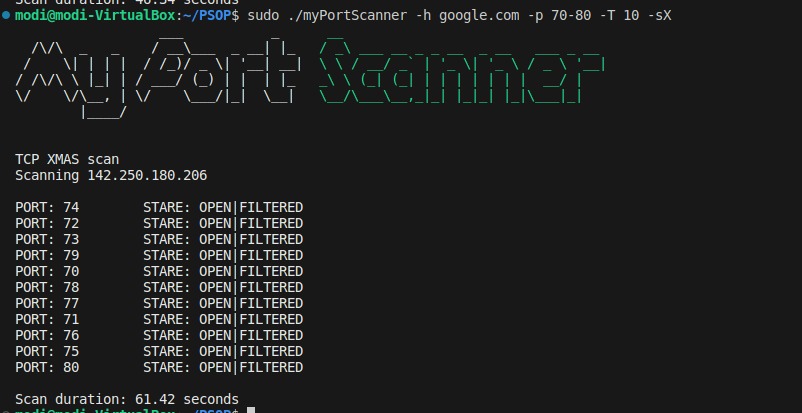
Input:

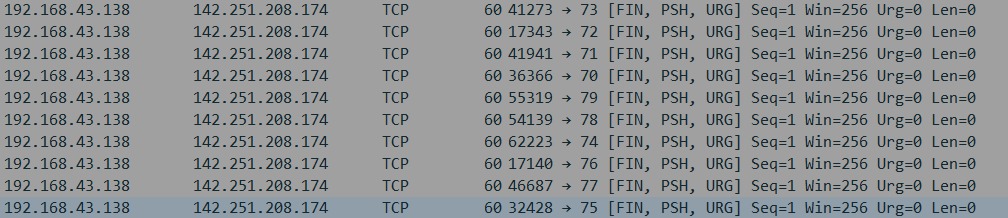
* Hostname: google.com
* Port: 70-80

Output:

**./myPortScanner:**







**Bibliografie:**

* **Resurse de studiu:**

1. <https://nmap.org/book/scan-methods.html>
2. <https://medium.com/@Aircon/nmap-advanced-port-scans-tryhackme-thm-ed3859a33eca>

* **Tratarea argumentelor:**

1. <https://www.gnu.org/software/libc/manual/html_node/Argp-Example-3.html>

* **getLocalIp:**

1. <https://www.binarytides.com/get-local-ip-c-linux/>

* **csum:**

1. https://www.gnu.org/software/libc/manual/html\_node/Argp-Example-3.html