

# Dokumentácia

2. PROJEKT PREDMETU IPK

HRUŠKA JOZEF (XHRUSK25)

## Table of Contents

<b>TEST .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
-------------------	-------------------------------------

## Použitá literatúra pri vývoji

Okrem webových stránok priložených priamo v zadaní projektu som využil pri tvorbe viaceré zdroje informácií:

- <https://www.tcpdump.org/pcap.html>  
<https://www.devdungeon.com/content/using-libpcap-c>  
- Podrobné informácie o práci s knižnicou *libpcap*.
- <https://www.tomicki.net/syn.flooding.php>  
- Problém SYN floodingu mi pomohol pri problémoch implementácie zasielania TCP SYN packetu.
- <https://opensourceforu.com/2015/03/a-guide-to-using-raw-sockets/>  
- Princíp práce s RAW socketmi

## Implementačné detaily

Náročnosť implementácie projektu som výrazne podcenil a preto som nestihol spracovať niektoré časti zadania.

### Hlavný program (ipk-scan.cpp):

Hlavné telo programu sa stará o celkovú správu chodu programu. Spracuje argumenty, skontroluje ich správnosť, skonvertuje ich hodnoty a podľa potreby vyvolá submoduly pre TCP a UDP scan portov.

#### *void **parseArguments();***

Metóda sa stará o správnu konverziu vstupných argumentov programu do pripravených štruktúr pre jednoduchšie využitie a neskoršie spracovanie.

#### *vector<int> **convertRange();***

Konverzia hodnôt parametrov špecifikujúcich testované porty. Pre rozpoznanie typu vstupných hodnôt využité regulárne výrazy.

### Modul – Scanner:

Abstraktná trieda ktorá slúži ako nadradená pre triedy jednotlivých skenerov. Implementuje metódu **startScan()**, uchováva vnútorné hodnoty skenerov a zastrešuje návratový typ jednotlivých skenov.

### Modul – TCP\_Scanner:

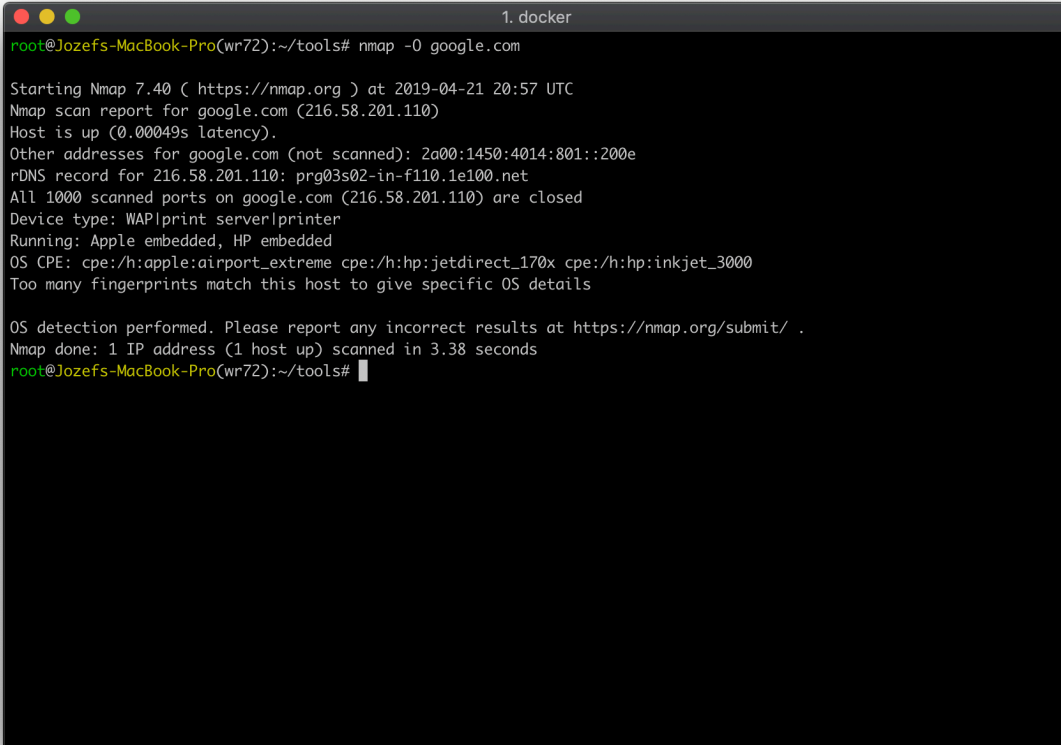
Trieda zapuzdrujúca implementáciu skenu TCP portov.

#### *Scanner::SCAN\_RESULT **startScan();***

Vytvorí príslušnú IP a TCP hlavičku a naplní ich dátami. Po spočítaní kontrolného súčtu inicializuje socket pre TCP komunikáciu a odošle SYN packet.

## Testovanie

Pre nedostatočnú implementáciu zadania bolo možné otestovať len iný nástroj.

A screenshot of a terminal window titled "1. docker". The prompt is "root@Jozefs-MacBook-Pro(wr72):~/tools#". The command entered is "nmap -O google.com". The output shows the Nmap 7.40 scan results for google.com (216.58.201.110), including host status, other addresses, rDNS record, scanned ports, device type, running OS, and OS CPE. The scan took 3.38 seconds.

```
1. docker
root@Jozefs-MacBook-Pro(wr72):~/tools# nmap -O google.com

Starting Nmap 7.40 ( https://nmap.org ) at 2019-04-21 20:57 UTC
Nmap scan report for google.com (216.58.201.110)
Host is up (0.00049s latency).
Other addresses for google.com (not scanned): 2a00:1450:4014:801::200e
rDNS record for 216.58.201.110: prg03s02-in-f110.1e100.net
All 1000 scanned ports on google.com (216.58.201.110) are closed
Device type: WAP|print server|printer
Running: Apple embedded, HP embedded
OS CPE: cpe:/h:apple:airport_extreme cpe:/h:hp:jetdirect_170x cpe:/h:hp:inkjet_3000
Too many fingerprints match this host to give specific OS details

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.38 seconds
root@Jozefs-MacBook-Pro(wr72):~/tools#
```

Obrázok 1 - Jednoduchý sken (NMAP)

## Kompilácia programu:

Kompilácia prebieha pomocou príkazu **,make’**. Makefile vytvorí spustiteľnú verziu programu, ktorá sa dá následne spustiť spôsobom, akým bolo požadované v zadaní.