PES UNIVERSITY

Computer Networks

[UE21CS251B]

PORT SCANNER

(PROJECT REPORT)

Group Members:

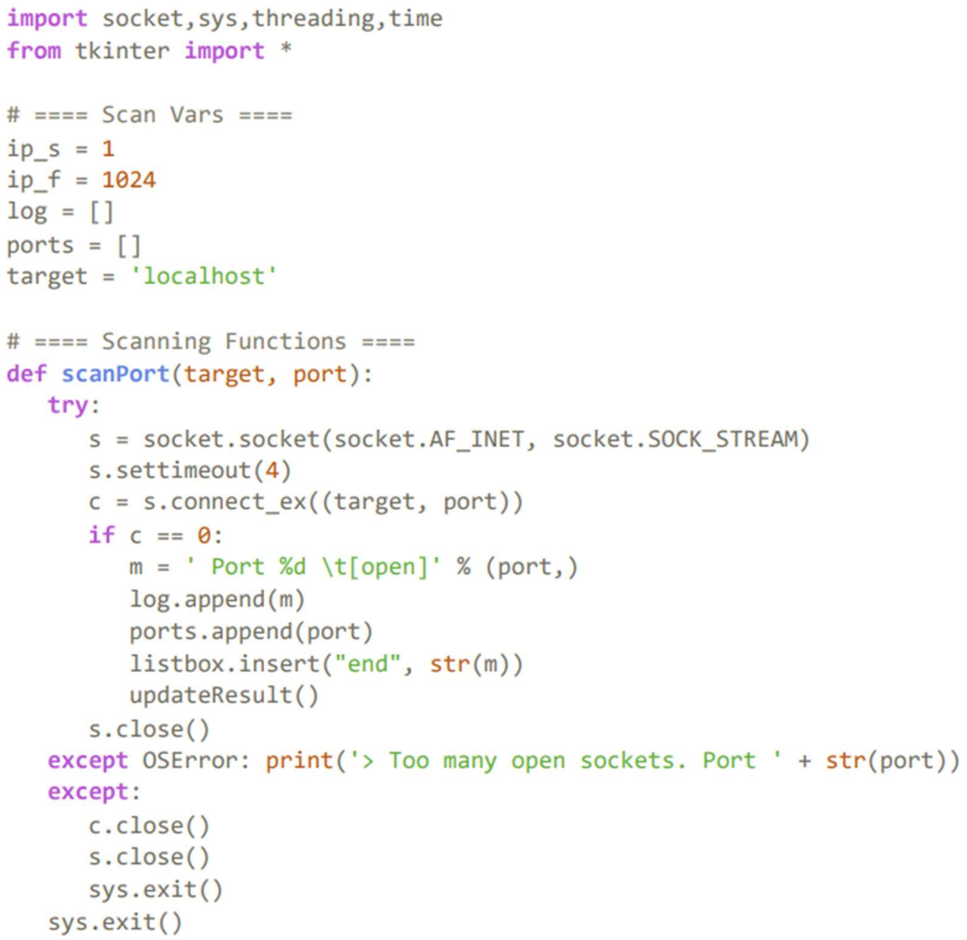
Shrey - PES2UG21CS500

Sujal - PES2UG21CS548

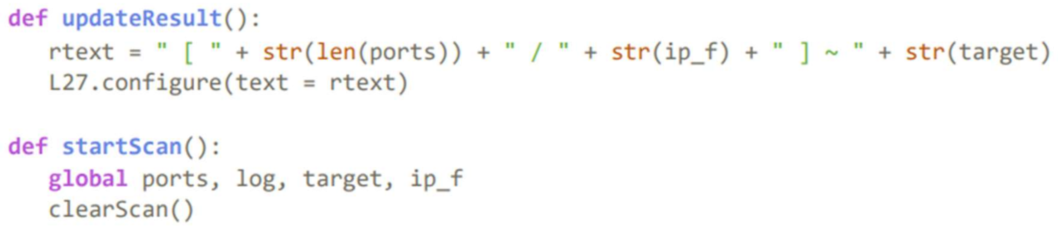
Sudhanva Samaga - PES2UG21CS544

- The Port Scanner project is a tool for identifying open ports on a target machine.
- The project is built using Python and the socket module.
- TCP protocol is used for scanning open ports.
- The project allows the user to specify a range of ports to scan or scan all ports on a given IP address.
- Results can be saved to a file for future reference.
- The project is useful for network administrators and cybersecurity experts.
- Future improvements can include additional scanning options and increased accuracy.
- The Port Scanner project successfully demonstrated the ability to scan for open ports on a target machine.

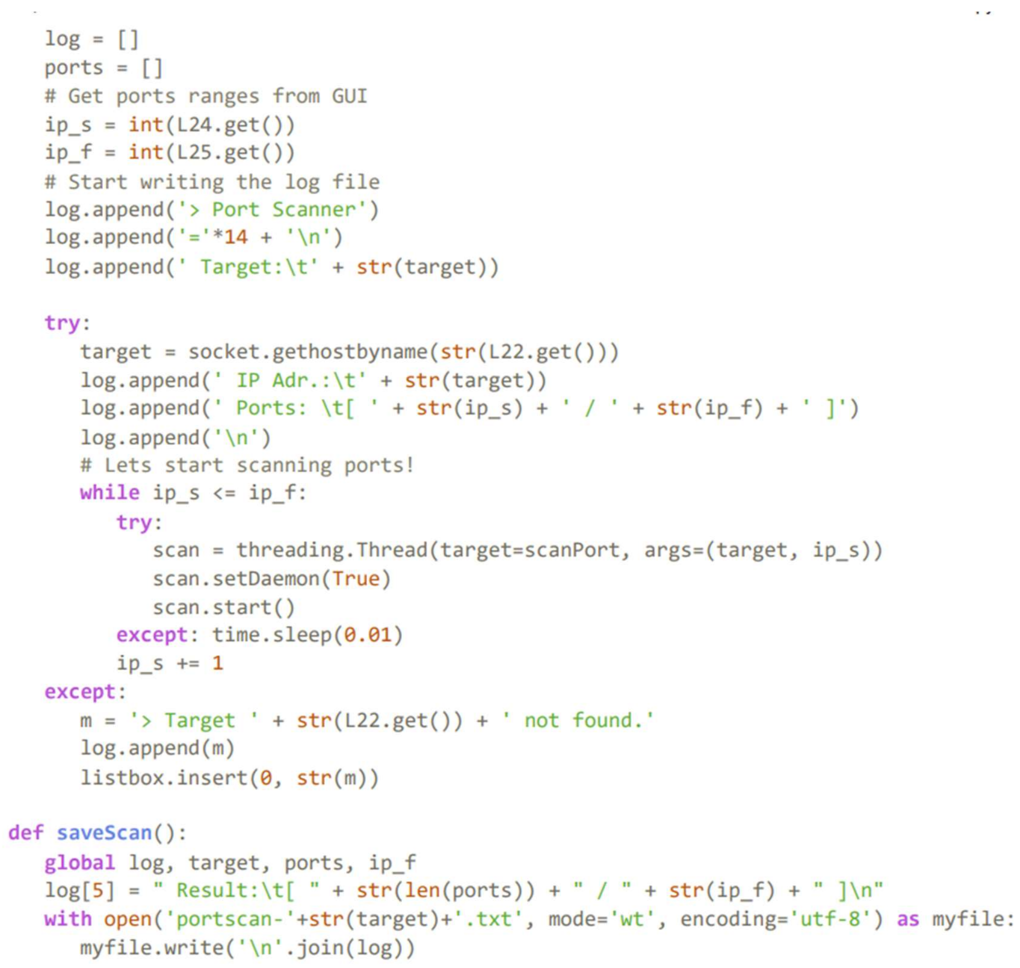
Our Code:



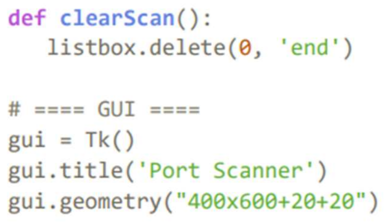
```
import socket,sys,threading,time
from tkinter import *

# ==== Scan Vars ====
ip_s = 1
ip_f = 1024
log = []
ports = []
target = 'localhost'

# ==== Scanning Functions ====
def scanPort(target, port):
   try:
      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
      s.settimeout(4)
      c = s.connect_ex((target, port))
      if c == 0:
         m = ' Port %d \t[open]' % (port,)
         log.append(m)
         ports.append(port)
         listbox.insert("end", str(m))
         updateResult()
      s.close()
   except OSError: print('> Too many open sockets. Port ' + str(port))
   except:
      c.close()
      s.close()
      sys.exit()
   sys.exit()
```



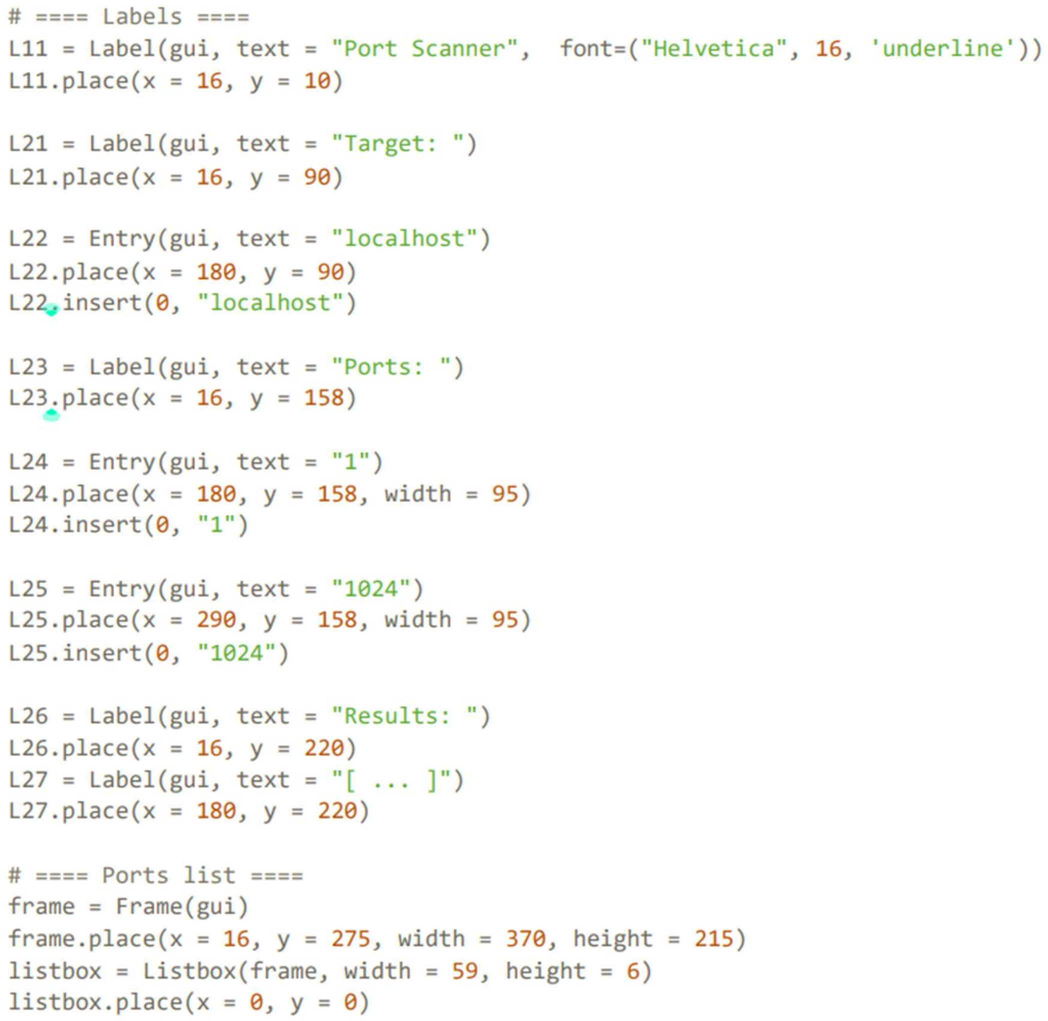
```
def updateResult():
   rtext = " [ " + str(len(ports)) + " / " + str(ip_f) + " ] ~ " + str(target)
   L27.configure(text = rtext)

def startScan():
   global ports, log, target, ip_f
   clearScan()
```



```
   log = []
   ports = []
   # Get ports ranges from GUI
   ip_s = int(L24.get())
   ip_f = int(L25.get())
   # Start writing the log file
   log.append('> Port Scanner')
   log.append('='*14 + '\n')
   log.append(' Target:\t' + str(target))

   try:
      target = socket.gethostbyname(str(L22.get()))
      log.append(' IP Adr.:\t' + str(target))
      log.append(' Ports: \t[ ' + str(ip_s) + ' / ' + str(ip_f) + ' ]')
      log.append('\n')
      # Lets start scanning ports!
      while ip_s <= ip_f:
         try:
            scan = threading.Thread(target=scanPort, args=(target, ip_s))
            scan.setDaemon(True)
            scan.start()
         except: time.sleep(0.01)
         ip_s += 1
   except:
      m = '> Target ' + str(L22.get()) + ' not found.'
      log.append(m)
      listbox.insert(0, str(m))

def saveScan():
   global log, target, ports, ip_f
   log[5] = " Result:\t[ " + str(len(ports)) + " / " + str(ip_f) + " ]\n"
   with open('portscan-'+str(target)+'.txt', mode='wt', encoding='utf-8') as myfile:
      myfile.write('\n'.join(log))
```

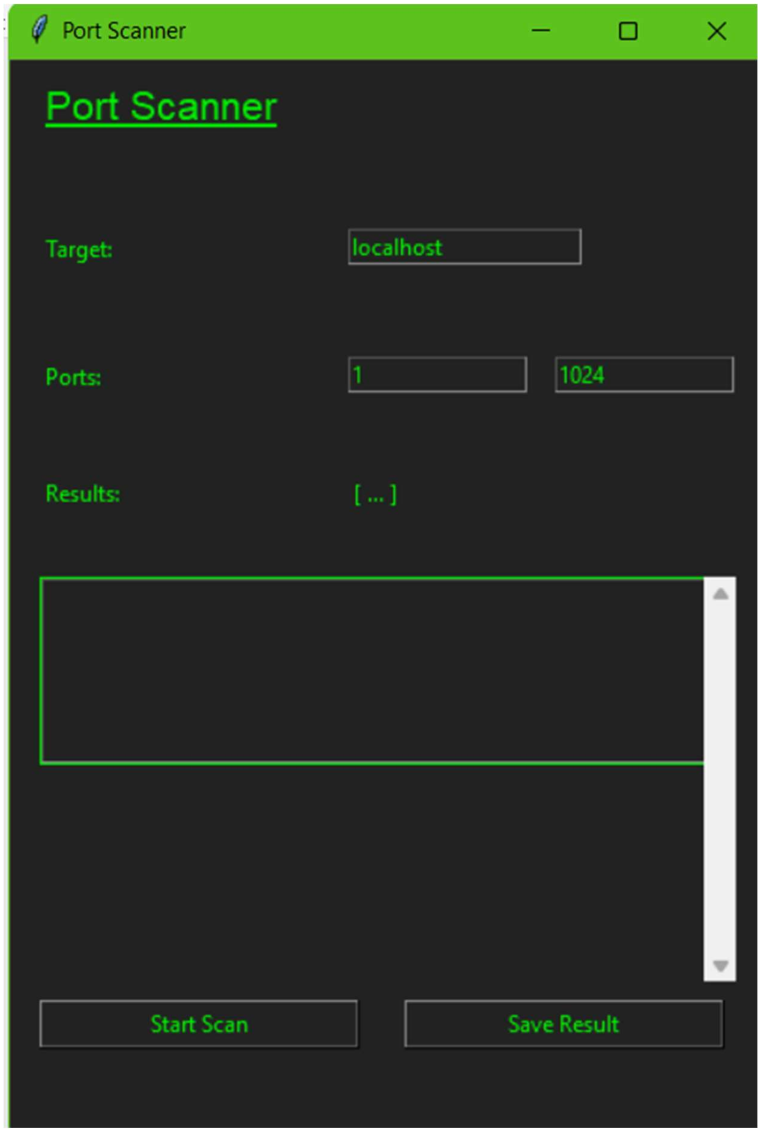


```
def clearScan():
   listbox.delete(0, 'end')

# ==== GUI ====
gui = Tk()
gui.title('Port Scanner')
gui.geometry("400x600+20+20")
```



```
# ==== Labels ====
L11 = Label(gui, text = "Port Scanner",  font=("Helvetica", 16, 'underline'))
L11.place(x = 16, y = 10)

L21 = Label(gui, text = "Target: ")
L21.place(x = 16, y = 90)

L22 = Entry(gui, text = "localhost")
L22.place(x = 180, y = 90)
L22.insert(0, "localhost")

L23 = Label(gui, text = "Ports: ")
L23.place(x = 16, y = 158)

L24 = Entry(gui, text = "1")
L24.place(x = 180, y = 158, width = 95)
L24.insert(0, "1")

L25 = Entry(gui, text = "1024")
L25.place(x = 290, y = 158, width = 95)
L25.insert(0, "1024")

L26 = Label(gui, text = "Results: ")
L26.place(x = 16, y = 220)
L27 = Label(gui, text = "[ ... ]")
L27.place(x = 180, y = 220)

# ==== Ports list ====
frame = Frame(gui)
frame.place(x = 16, y = 275, width = 370, height = 215)
listbox = Listbox(frame, width = 59, height = 6)
listbox.place(x = 0, y = 0)
```

OUTPUT:



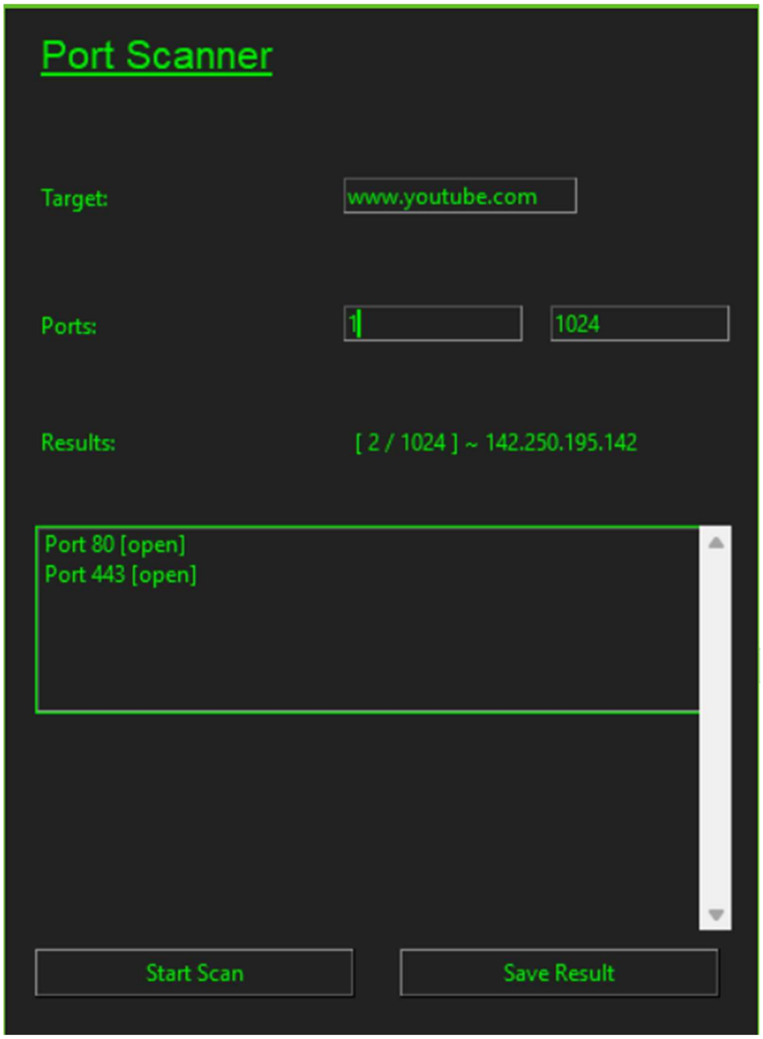
Port Scanner

Target: localhost

Ports: 1 1024

Results: [...]

Ports Opened in www.youtube.com



Port Scanner

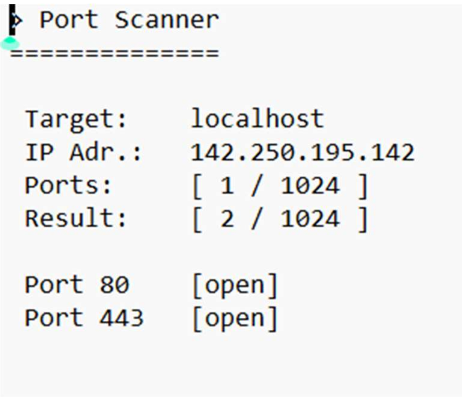
Target: www.youtube.com

Ports: 1 1024

Results: [2 / 1024] ~ 142.250.195.142

```
Port 80 [open]
Port 443 [open]
```

We can even save the above for future references:



```
> Port Scanner
==============

 Target:     localhost
 IP Adr.:    142.250.195.142
 Ports:      [ 1 / 1024 ]
 Result:     [ 2 / 1024 ]

 Port 80     [open]
 Port 443    [open]
```