1 simple port scanner— scanning port 80

```python
#!/bin/python3

import socket # Socket Library
from IPy import IP # IP Library

ipaddress = input(' [+} Enter target to scan: ') # Parameter to enable us to scan port 80
port = 80
try:
    sock = socket.socket() # Parameter to connect to internnet
    sock.connect ((ipaddress, port)) #  Try to connect the IP address of the port
    print('[+] port 80 is open')  # Print open if the port is open
except:
    print('[-] port 80 is closeed') # Print close if the port is closed
```

Two libraries was imported i.e Socket and IPy. We declared two variables i.e Port and IP and the first variables enables us to type IPaddress that we want to scan while PORT was hard coded to PORT 80

We imported two libraries

1. Socket
2. From IPy import IP

Declare 2 Variables

1. Ipaddress - type in the ipaddress for the purpose of scan
2. Port - hard coded port 80

Using “sock” we use try and except command to connect socket

2 Portscanner 2 - scanning multiple ipaddress

```python
#!/bin/python3

import socket # Socket Library
from IPy import IP # IP Library



def scan_port(ipaddress, port): # Function defining two variable
    try:
        sock = socket.socket() # Parameter to connect the internet
        sock.sttimeout(0.5) # Parameter that set the time for each scan of the port
        sock.connect ((ipaddress, port)) # Parameter that connect ipaddress and port
        print('[+] port' + str(port) + 'is open') # print a defined number of open port from the target
    except:
        print('[-] port' + str(port) + 'is closeed') # print a defined number of close port from the target

ipaddress = input('[+] Enter Target to scan: ') # The targetted ip address
for port in range(1,25): # for loop that allow the program to scan multipe port
    scan_port(ipaddress, port) # Scan multiple port using multiply Scan_port with two parameter
```

The two variable we have is ipaddress and scan_port. The essence of scan_port is to enable multiple scanning

1. Redefining variable 2 port with a for loop expanding the range to 25 and beyond
2. Created a function called scan_port with (ipaddress, port) as parameter
3. Modify the try and bexcept statement
4. Added sock.settimeout to regulate the scanning time

3 PORTSCANNER 3 : scanning without inputting ip address

```
Open
#!bin/python3

import socket  # Socket Library
from IPy import IP  # IP Library



def check_ip(ip): # Function defining converted ip
    try:
        IP(ip) # check ip
        return(ip) # return to the host if there is value error
    except ValueError:
        return socket.gethostbyname(ip) # the website called host

def scan_port(ipaddress, port): # Function defining two veriables
    try:
        sock = socket.socket()  # parameter to connect to internet
        sock.sttimeout(0.5)    # Parameter that the amount of time used to sacn each port
        sock.connect ((ipaddress, port))
        print('[+] port' + str(port) + 'is open')
    except:
        print('[-] port' + str(port) + 'is closeed')

ipaddress = input('[+] Enter Target to scan: ') # To enable us scan website address directly
converted_ip = check_ip(ipaddress) # Declaring a variable check_ip with parameter ipaddress

for port in range(1,50): # Defined number of converted ip port
    scan_port(converted_ip, port)

```

1. To enable us scan a domain name and ipaddress we modify the parameter ipaddress in function scan_port
2. Introduced converted_ip = check_ip(ipaddress) with a single parameter ipaddress
3. We define the function check_ip with single parameter (ip)
4. Calling IPy library and using try and excpt statement to allow the program check_ip or get gethostbyname

4 PORTSCANNER 4: Scanning with multiple domain name

```python
#!/bin/python3

import socket
from IPy import IP

def scan(target):
    converted_ip = check_ip(target)
    print('\n' + '[-_0 Scanning Target]' + str(target))
    for port in range(1,50):
        scan_port(converted_ip,port)

def check_ip(ip):
    try:
        IP(ip)
        return(ip)
    except ValueError:
        return socket.gethostbyname(ip)

def get_banner(s):
    return s.recv(1024)

def scan_port(ipaddress, port):
    try:
        sock = socket.socket()
        sock.settimeout(0.5)
        sock.connect ((ipaddress, port))
        try:
            banner = get_banner(sock)
            print(' [+] open port ' + str(port) + ' : ' + str(banner.decode().strip('\n')))
        except:
            print(' [+] open port ' + str(port))
    except:
        pass

targets = input('[+] Enter Target/s to scan(split multiple target with ,:) ')

if ',' in targets:
    for ip_add in targets.split(','):
        scan(ip_add.strip(' '))

else:
    scan(targets)
```

1. To enable us scan multiple domain name and ipaddress we change ipadress variable to target
2. We created an if statement to enable us list domain name with, “‘,” separating the input and scan
3. We define a scan statement and replace ipadress with target
4. We define the scan function

PORT SCANNER BANNER GRABBER

1. We define the banner grabber function and the variable
2. We modify the try and except statement

```python
def get_banner(s):
    return s.recv(1024)
```

```python
def scan_port(ipaddress, port):
    try:
        sock = socket.socket()
        sock.settimeout(0.5)
        sock.connect ((ipaddress, port))
        try:
            banner = get_banner(sock)
            print(' [+] open port ' + str(port) + ' : ' + str(banner.decode().strip('\n')))
        except:
            print(' [+] open port ' + str(port))
    except:
        pass
```