

*A Project Report on*

# **OP - Scan (An Open Port Scanner)**

(Project for Cyber Security - CSE 4026)

*Submitted by*

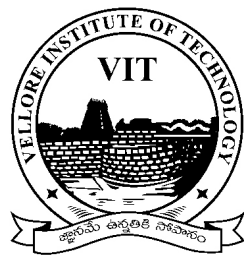
**Nishit Verma**

(Registration-19BCN7050)

*on*

**31 December 2021**

**Slot : F**



**VIT<sup>®</sup>**  

---

**AP**

**School of Computer Science and Engineering**

**VIT-AP University**

**Andhra Pradesh**

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Objective</b>	<b>4</b>
<b>3</b>	<b>Methodology</b>	<b>5</b>
<b>4</b>	<b>Usage</b>	<b>9</b>
4.1	Installation . . . . .	9
4.2	How to Use . . . . .	10
<b>5</b>	<b>Conclusion</b>	<b>13</b>
<b>6</b>	<b>Appendix</b>	<b>14</b>

# 1 Introduction

- OP - Scan is an open sourced open port scanner.
- Port Scanning is the next phase in an Ethical Hacking Reconnaissance Plan and follows on from the foot printing phase.
- It assists us in identifying the services the target is running. For example, if a port scan reveals port 80 TCP is open we know there is a web service running on the target device.
- This tool helps us to find such open ports without putting in much effort and retrieves the result in a very short period of time, helping us to save a lot of time while performing recon.

```
(root@kali)~/OP-Scan] connect(ip, port)
# python3 OP-Scan.py --url www.hackerrank.com --start 1 --end 1000

OP-SCAN

Enter URL/IP : www.hackerrank.com
Would you like to scan 'www.hackerrank.com' (y/n) -- y
Would you like to scan all (1-65535) ports (y/n) -- n
Enter Starting Port : 1
Enter Ending Port : 1000
Scan Started at : 2021-12-28 00:06:46.488146

PORT    STATUS  SERVICE
80      Open    http
443     Open    https

Scan Successfully Ended at : 2021-12-28 00:08:35.782186
Scan Duration : 0h - 1m - 49s
The Target IP : 173.223.50.43 has 2 open ports

(root@kali)~/OP-Scan]
```

Figure 1: Open Ports Found on Hacker Rank

## 2 Objective

- To find the Open ports on a particular target website or IP address.
- To speed up the foot printing phase and save the time of pentesters.
- To help the bug hunters and security advisors find out open ports, test them and filter them.

### 3 Methodology

- The OP - Scan tool is written in python and can be broken into 3 parts.

#### 1. Import

```
import pyfiglet
from datetime import datetime
import socket
```

- The pyfiglet takes ASCII text and renders it in ASCII art fonts.
- From the default datetime package, we import date and time to note the start and end time of scan.
- Finally we import socket to implement socket programming in python.

#### 2. Target data and Scan Input

```
target = input("Enter URL/IP : ")
q1 = input("Would you like to scan '"+target+"' (y/n) -- ")
if q1[0] == "N" or q1[0] == "n":
    print("\nSession Terminated !!!")
    exit()
else:
    Sport = 1
    Eport = 65535
    q2 = input("Would you like to scan all (1-65535) ports (y/n) -- ")
    if q2[0] == "Y" or q2[0] == "y":
        Stime=datetime.now()
        print("Scan Started at : "+str(Stime))
    else:
        Sport = int(input("Enter Starting Port : "))
        Eport = int(input("Enter Ending Port : "))
        Stime=datetime.now()
        print("Scan Started at : "+str(Stime))
```

- Here the tool will ask the user to input the URL or IP address of the target.
- Then the tool will ask to confirm that the input provided is correct and they want to continue as it is illegal to scan or monitor someone's network or server.
- On agreeing to continue, the tool will ask them to choose if they want to scan all the 1-65535 ports. As in general for quick scans, we only scan for particular port numbers or a particular range of ports.
- If they agree to scan all, the tool will proceed with the scanning of all the 65535 ports else it will ask them to enter the starting and ending port numbers.  
(Same port numbers can be entered in both the sections if only one port is needed to be scanned)

### 3. Output

```

        ip = socket.gethostbyname(target)
except socket.gaierror:
print("\nHostname Could Not Be Resolved !!!!")
exit()
c = 0
print("PORT\tSTATUS\tSERVICE")
for port in range(Sport,Eport+1):
i = 0
try:
sck = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sck.settimeout(0.1)
except socket.error:
        print("\nServer not responding !!!!")
        exit()
try:
sck.connect((ip, port))
sck.settimeout(None)
i = 1
except Exception as e:
i = 0;

```

```

if i == 1:
    ser = socket.getservbyport(port)
    print(str(port)+"\tOpen\t"+ser)
    c = c+1
if c == 0:
    print("\nNo Open Ports Found")
    Etime = datetime.now()
    duration = Etime - Stime
    duration_in_s = duration.total_seconds()
    hours = divmod(duration_in_s, 3600)
    minutes = divmod(hours[1], 60)
    seconds = divmod(minutes[1], 1)
    print("\nScan Successfully Ended at : "+str(Etime))
    print("Scan Duration : "+str(int(hours[0]))+"h - "+str(int(minutes[0]))+"m - "+str(int(seconds[0]))+"s")
    print("The Target IP : "+str(ip)+" has "+str(c)+" open ports")

```

- Now in the first step, the IP address of the target will be acquired.
- Then the tool will run a loop of from the starting port to ending port so that it can verify each port.
- Now the tool will create the socket and set the timeout to 0.1 sec so that the scan won't be delayed due to failed connection attempts.
- Finally the tool will try to connect the socket to the server of the provided target IP with the respective port number one by one from the loop, If the connection fails, the process continues without any output but if the connection gets established successfully, the tool will report that the particular port is open and will display its service along with that.
- Kindly refer the **Appendix** for the complete python code.

- The key concept behind the above python script is that when we attempt to connect to a server using TCP sockets, we require TCP 3 way handshake.

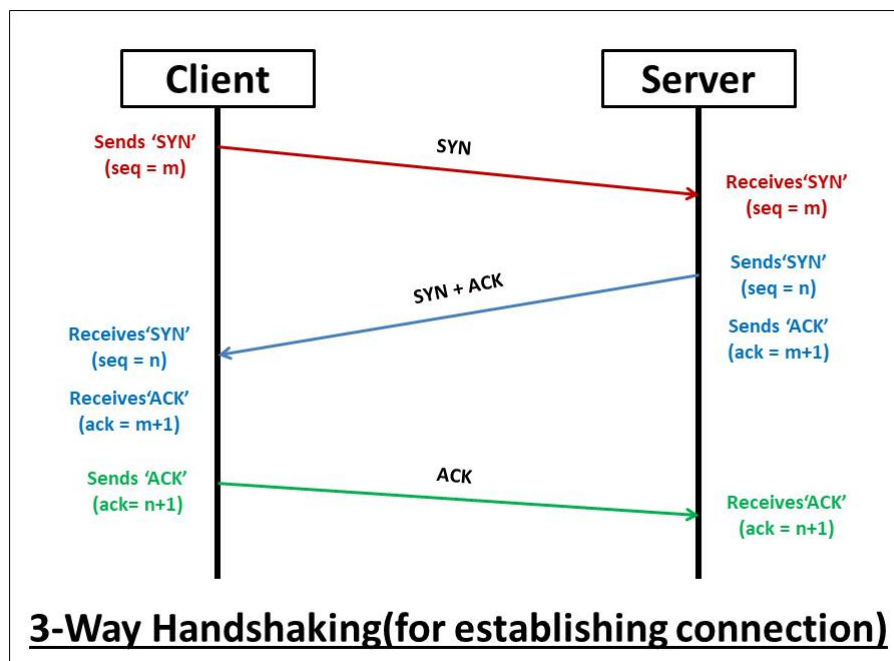


Figure 2: TCP 3 Way Handshake Process

- Now if the port is busy or inactive or filtered it will not perform the 3 way handshake and if the port is open, the server will perform the 3 way handshake.
- So, based on that simple logic the above python script is coded. In the script, as we discussed if the connection is successful then we print the port is open because the 3 way handshake has been successful else we handle the error and in turn we skip that port because it's either closed, filtered or busy.



## 4 Usage

### 4.1 Installation

- The tool OP - Scan is an open sourced tool so you can easily download that from Github or can clone it from Github directly into the respective folder or directory using the below code snippet.

```
git clone https://github.com/Nishit3479/OP-Scan.git
```

A terminal window screenshot showing the process of cloning a repository. The prompt is (root@kali) - [~/Demo]. The user enters 'ls' and then 'git clone https://github.com/Nishit3479/OP-Scan.git'. The output shows the cloning progress: 'Cloning into 'OP-Scan'...', 'remote: Enumerating objects: 10, done.', 'remote: Counting objects: 100% (10/10), done.', 'remote: Compressing objects: 100% (5/5), done.', 'remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Receiving objects: 100% (10/10), done.'. After the cloning is complete, the user enters 'ls' again, and the output shows 'OP-Scan' as a new directory in the current path.

```
(root@kali) - [~/Demo]
# ls
(root@kali) - [~/Demo]
# git clone https://github.com/Nishit3479/OP-Scan.git
Cloning into 'OP-Scan'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
(root@kali) - [~/Demo]
# ls
OP-Scan
(root@kali) - [~/Demo]
#
```

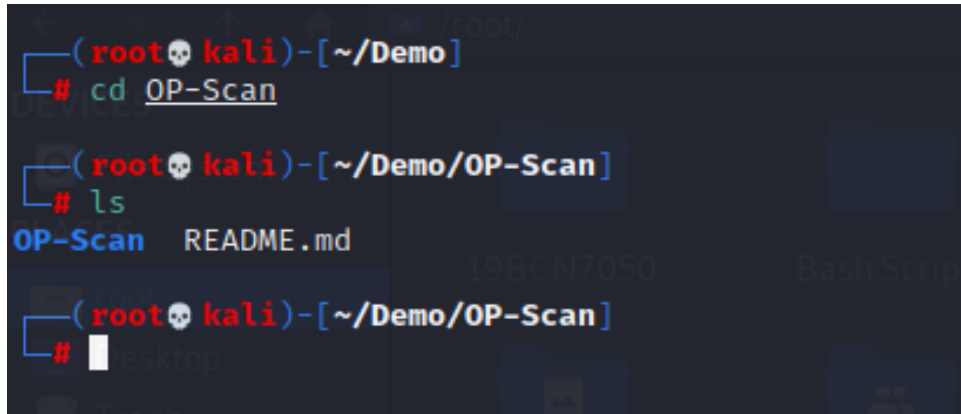
Figure 3: Cloning OP - Scan from Github

- The **pyfiglet** package of python needs to be installed prior usage for proper implementation of the tool. You can install it by entering the following command.

```
pip install pyfiglet
```

## 4.2 How to Use

1. Change the directory from the download directory to OP - Scan.



```
(root👤kali)-[~/Demo]
# cd OP-Scan

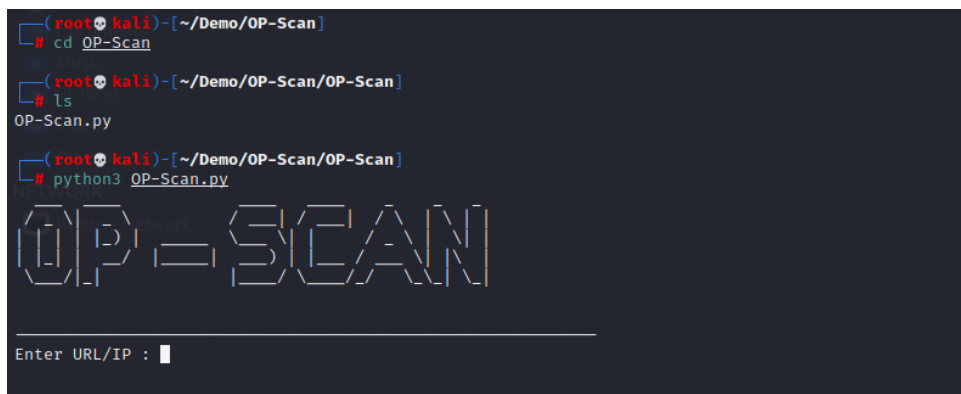
(root👤kali)-[~/Demo/OP-Scan]
# ls
OP-Scan  README.md

(root👤kali)-[~/Demo/OP-Scan]
#
```

Figure 4: Changing directory from Demo to OP - Scan

2. Now repeat the same process as above to change the directory from OP - Scan to OP - Scan folder that is inside that and then to execute the tool just enter the following command.

```
python3 OP-Scan.py
```



```
(root👤kali)-[~/Demo/OP-Scan]
# cd OP-Scan

(root👤kali)-[~/Demo/OP-Scan/OP-Scan]
# ls
OP-Scan.py

(root👤kali)-[~/Demo/OP-Scan/OP-Scan]
# python3 OP-Scan.py

OP-SCAN

Enter URL/IP : 
```

Figure 5: Executing OP-Scan.py

3. Now enter the target IP Address or URL, then agree to scan. Next you will be asked whether you want to scan all the ports i.e., from port - 1 to port - 65535 or you want to scan a particular range of ports.

Here in this demo we will go with the range of ports from port - 1 to port - 1000 and the target URL will be `www.hackerrank.com`.

Now on clicking Enter, the tool will start to scan for the open ports.

```
(root@kali)~[~/Demo/OP-Scan/OP-Scan]
# python3 OP-Scan.py

OP-SCAN

Enter URL/IP : www.hackerrank.com
Would you like to scan 'www.hackerrank.com' (y/n) -- y
Would you like to scan all (1-65535) ports (y/n) -- n
Enter Starting Port : 1
Enter Ending Port : 1000
Scan Started at : 2021-12-31 05:51:03.006867

PORT    STATUS  SERVICE
|
```

Figure 6: Entering data Starting the scan

4. Finally within a few moments, the tool completes the scanning and displays all the open ports found with their respective service names. In the demo, we found 2 open ports i.e., port - 80 and port - 443.

```
(root@kali)-[~/Demo/OP-Scan/OP-Scan]
# python3 OP-Scan.py

OP-SCAN

Enter URL/IP : www.hackerrank.com
Would you like to scan 'www.hackerrank.com' (y/n) -- y
Would you like to scan all (1-65535) ports (y/n) -- n
Enter Starting Port : 1
Enter Ending Port : 1000
Scan Started at : 2021-12-31 05:51:03.006867

PORT    STATUS  SERVICE
80      Open    http
443     Open    https

Scan Successfully Ended at : 2021-12-31 05:52:52.690449
Scan Duration : 0h - 1m - 49s
The Target IP : 104.120.93.79 has 2 open ports

(root@kali)-[~/Demo/OP-Scan/OP-Scan]
#
```

Figure 7: Result of the Scan - Open ports found

## 5 Conclusion

- Based on the above demonstration, we can conclude that this tool can prove to be useful for people such as pentesters who would like to use a simple tool to find the open ports in the target URL or IP address's server in a small period of time.
- By reading the basic structure of the tool, we can also understand that the tool is really fast for smaller number of ports in comparison to larger number of ports when we compare it with other open port scanners like nmap.

## 6 Appendix

### OP-Scan.py

```
import pyfiglet
from datetime import datetime
import socket

op_scan = pyfiglet.figlet_format("OP - SCAN")
print(op_scan)
print("-" * 60)

try:
    target = input("Enter URL/IP : ")
    q1 = input("Would you like to scan '"+target+"' (y/n) -- ")
    if q1[0] == "N" or q1[0] == "n":
        print("\nSession Terminated !!!")
        exit()
    else:
        Sport = 1
        Eport = 65535
        q2 = input("Would you like to scan all (1-65535) ports (y/n) -- ")
        if q2[0] == "Y" or q2[0] == "y":
            Stime=datetime.now()
            print("Scan Started at : "+str(Stime))
        else:
            Sport = int(input("Enter Starting Port : "))
            Eport = int(input("Enter Ending Port : "))
            Stime=datetime.now()
            print("Scan Started at : "+str(Stime))
        print("-" * 60)
    try:
        ip = socket.gethostbyname(target)
    except socket.gaierror:
```

```

        print("\nHostname Could Not Be Resolved !!!!")
        exit()

c = 0
print("PORT\tSTATUS\tSERVICE")
for port in range(Sport,Eport+1):
    i = 0
    try:
        sck = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sck.settimeout(0.1)
    except socket.error:
        print("\nServer not responding !!!!")
        exit()

    try:
        sck.connect((ip, port))
        sck.settimeout(None)
        i = 1
    except Exception as e:
        i = 0;
    if i == 1:
        ser = socket.getservbyport(port)
        print(str(port)+"\tOpen\t"+ser)
        c = c+1
    if c == 0:
        print("\nNo Open Ports Found")
    Etime = datetime.now()
    duration = Etime - Stime
    duration_in_s = duration.total_seconds()
    hours    = divmod(duration_in_s, 3600)
    minutes  = divmod(hours[1], 60)
    seconds  = divmod(minutes[1], 1)
    print("\nScan Successfully Ended at : "+str(Etime))
    print("Scan Duration : "+str(int(hours[0]))+"h - "+str(int(minutes[0]))+"m - "+str(int(seconds[0]))+"s")

```

```
"m - "+str(int(seconds[0]))+"s")
print("The Target IP : "+str(ip)+" has "+str(c)+" open ports")
except KeyboardInterrupt:
    print("\nSession Terminated !!!")
    exit()
```

---