

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش تمرین کد نویسی اول

نام کامل نویسنده:

مرتضی دامغانی نوری

(۹۶۲۵۸۰۱)

نام استاد درس:

استاد شهریار

آبان ماه هزار و چهار صد

بخش اول:

گرفتن ping از آیپی خاص:

```
import subprocess
while True:
    ping_input = input("Enter the IP address or the URL you want to ping or enter 'exit' to exit: ")
    if ping_input == "exit":
        break
    ping_result = subprocess.run("ping " + ping_input)
    print(str(ping_result) + "\n")
```

در بالا تصویری از کد نوشته شده برای این قسمت آورده شده است و همان طور که در تصویر بالا قابل مشاهده است، برای انجام این قسمت از کتابخانه ی subprocess در python استفاده شده است و ابتدا از کاربر آیپی مورد نظر یا URL مورد نظر گرفته می شود و سپس به عنوان ورودی به کتابخانه ی subprocess داده می شود و خروجی دریافت شده توسط این کتابخانه در کنسول نمایش داده می شود. کد مربوط به این قسمت در فولدر Ping و در ping.py قابل مشاهده است و همچنین خروجی های این قسمت در فایل result_[ping].txt قرار داده شده است.

برای تست این قسمت یک بار www.google.com و یک بار آیپی مربوط به سایت گوگل که عبارت است از 216.58.209.132 به برنامه داده شد و تصویری از این خروجی در ادامه آورده شده است و این خروجی در فایل result_[ping].txt نیز قرار داده شده است.

```

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "E:/MortezaDamghaniNouri/MyCodes/Python Codes/Information Security Programming Assignment 1/ping.py"
Enter the IP address or the URL you want to ping or enter 'exit' to exit: www.google.com

Pinging www.google.com [216.58.209.132] with 32 bytes of data:
Reply from 216.58.209.132: bytes=32 time=62ms TTL=112
Reply from 216.58.209.132: bytes=32 time=61ms TTL=112
Reply from 216.58.209.132: bytes=32 time=61ms TTL=112
Reply from 216.58.209.132: bytes=32 time=61ms TTL=112

Ping statistics for 216.58.209.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 61ms, Maximum = 62ms, Average = 61ms
CompletedProcess(args='ping www.google.com', returncode=0)

Enter the IP address or the URL you want to ping or enter 'exit' to exit: 216.58.209.132

Pinging 216.58.209.132 with 32 bytes of data:
Reply from 216.58.209.132: bytes=32 time=62ms TTL=112
Reply from 216.58.209.132: bytes=32 time=62ms TTL=112
Reply from 216.58.209.132: bytes=32 time=62ms TTL=112
Reply from 216.58.209.132: bytes=32 time=63ms TTL=112

Ping statistics for 216.58.209.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 62ms, Maximum = 63ms, Average = 62ms
CompletedProcess(args='ping 216.58.209.132', returncode=0)

```

اسکن یک محدوده آیپی و یافتن هاست های فعال:

```

import subprocess

while True:
    network_address = input("Enter the network address or enter 'exit' to exit: ")
    if network_address == "exit":
        break
    starting_number = int(input("Enter the starting number: "))
    last_number = int(input("Enter the last number: "))
    print("Processing...")
    network_address_list = list(network_address)
    while network_address_list[len(network_address_list) - 1] != ".":
        network_address_list.pop(len(network_address_list) - 1)
    network_address = ""
    for num in network_address_list:
        network_address += num
    i = starting_number
    while i <= last_number:
        popen_result = subprocess.Popen("ping -l 8 -n 2 " + network_address + str(i), stdout=subprocess.PIPE)
        popen_data = popen_result.communicate()[0]
        return_code = popen_result.returncode
        if str(return_code) == "0":
            print(network_address + str(i) + " " + "--> " + "Live")
        i += 1
    print("Scanning completed")

```

همان طور که در تصویر بالا قابل مشاهده است، کد مربوط به این بخش به این شکل نوشته شده است که حلقه ی `while` بی نهایتی قرار داده شده است که تا زمانی که ورودی ای به غیر از `exit` توسط کاربر وارد شود، ادامه پیدا می کند. در این حلقه ابتدا آدرس آیپی مورد نظر از کاربر دریافت می شود و سپس دو عدد دیگر نیز از کاربر دریافت می شود که به ترتیب مشخص کننده ی سه رقم آخر آدرس آیپی شروع محدوده ی مورد نظر و سه رقم آخر آدرس آیپی پایانی در محدوده ی مورد نظر است. شیوه ی بررسی مجموعه هاست های فعال در محدوده ی وارد شده به این شکل است که از تک تک آدرس های آیپی موجود در این محدوده `ping` گرفته می شود. شیوه ی پینگ گرفتن نیز مانند بخش قبل (گرفتن `ping` از آیپی خاص) است. بعد از آنکه از آدرس آیپی مورد نظر به کمک کتابخانه ی `subprocess` در پایتون `ping` گرفته شد `return code` آن بررسی می شود و اگر `return code` برابر با صفر بود نتیجه گرفته می شود که هاست مورد نظر فعال است و اگر `return code` مورد نظر عددی غیر از صفر بود نتیجه گرفته می شود که هاست مورد نظر غیر فعال است. برای تست کد مربوط به این قسمت، همان طور که در تصویر زیر قابل مشاهده است، آیپی `89.43.3.0` برای جستجوی محدوده آیپی `89.43.3.60` تا `89.43.3.70` به برنامه داده شده است.

```
Enter the network address or enter 'exit' to exit: 89.43.3.0
Enter the starting number: 60
Enter the last number: 70
```

در ادامه تصویری از خروجی مربوط به این ورودی آورده شده است:

```
Enter the network address or enter 'exit' to exit: 89.43.3.0
Enter the starting number: 60
Enter the last number: 70
Processing...
89.43.3.66 --> Live
89.43.3.67 --> Live
89.43.3.68 --> Live
89.43.3.69 --> Live
89.43.3.70 --> Live
Scanning completed
```

همان طور که در تصویر بالا قابل مشاهده است، در محدوده ی مشخص شده تنها آدرس هایی که سه رقم آخر آن ها ۶۶، ۶۷، ۶۸، ۶۹ و ۷۰ است، فعال اند.

کد مربوط به این بخش در فولدر Finding Alive Hosts و در فایل Finding Alive Host.py قرار داده شده است و خروجی حاصل از اسکن محدوده آییی ای که در تصویر بالا قابل مشاهده است نیز در فایل result_[Finding Alive Hosts].txt قرار داده شده است.

اسکن پورت های باز یک هاست فعال:

```
import nmap

while True:
    host_ip = input("Enter the host IP you want to scan its ports or enter 'exit' to exit: ")
    if host_ip == "exit":
        break
    start_port_number = input("Enter the start port number: ")
    last_port_number = input("Enter the last port number: ")
    print("Processing...")
    nm = nmap.PortScanner()
    nm.scan(host_ip, start_port_number + "-" + last_port_number)
    for host in nm.all_hosts():
        print('-----')
        for proto in nm[host].all_protocols():
            print('-----')
            print("Protocol: " + str(proto))
            ports_list = nm[host][proto].keys()
            for port in ports_list:
                print('Port : %s\tState : %s' % (port, nm[host][proto][port]['state']))
    print("Ports scanning completed")
```

همان طور که در تصویر بالا قابل مشاهده است، نحوه ی پیاده سازی این بخش به این شکل است که ابتدا آدرس آیپی هاستی که کاربر قصد بررسی پورت های آن را دارد از او گرفته می شود و پس از آن پورت های مورد نظر دریافت می شوند. سپس به کمک تابع `PortScanner` در کتابخانه ی `nmap` در زبان پایتون، پورت های مورد نظر آدرس آیپی وارد شده بررسی می شوند. این تابع یک دیکشنری باز می گرداند که `state` تمام پورت ها در محدوده ی وارد شده در آن مشخص شده است. سپس با انجام یک پیمایش بر روی این دیکشنری می توان وضعیت پورت های مورد نظر را پیدا کرد و به کاربر نمایش داد.

برای تست کردن این بخش پورت های ۵ تا ۹ از آدرس آیپی 89.43.3.170 بررسی شد و خروجی آن در تصویر زیر نشان داده شده است:

```
Port : 5    State : filtered
Port : 6    State : filtered
Port : 7    State : open
Port : 8    State : filtered
Port : 9    State : open
```

کد این بخش در داخل فولدر Finding Open Ports of a Host و در داخل فایل Finding Open Ports of a Host.py قرار داده شده است و همچنین خروجی مربوط به ورودی بالا در فایل result[finding open ports of a host] قرار داده شده است.

در فولدر Codes، یک فایل پایتونی با عنوان Part1 Complete.py قرار داده شده است که در آن تمام سه قسمت مربوط به بخش اول در قالب یک برنامه توسعه داده شده است.

بخش دوم:

ابزار nmap:

ابزاری است که از آن عمدتاً برای یافتن پورت های فعال یک آدرس آیپی استفاده می شود. این ابزار بر روی دو سیستم عامل ویندوز و لینوکس قابل استفاده است و برای انجام این قسمت از تمرین نسخه ی ویندوز آن نصب شد و سپس همان ورودی ای که به بخش سوم از بخش اول پروژه داده شد، در اختیار این نرم افزار نیز قرار گرفت. این ورودی عبارت است از مشخص کردن وضعیت پورت های ۵ تا ۹ از هاستی با آدرس آیپی 89.43.3.170. خروجی این نرم افزار در تصویر زیر قابل مشاهده است:

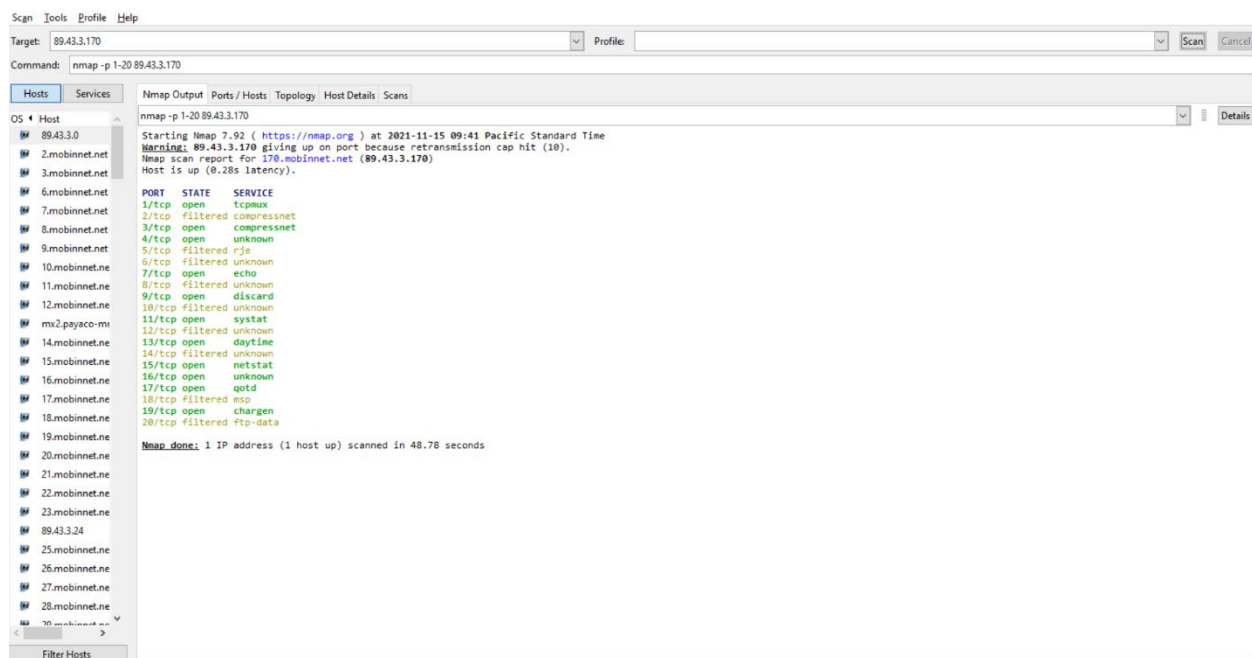
```

5/tcp  filtered rje
6/tcp  filtered unknown
7/tcp  open    echo
8/tcp  filtered unknown
9/tcp  open    discard

```

با بررسی این خروجی و مقایسه ی آن با خروجی کد نوشته شده می توان نتیجه گرفت که کد نوشته شده به درستی عمل کرده است.

در ادامه تصویری از محیط این نرم افزار به همراه اجرای یک دستور دیگر در آن آورده شده است:



در تصویر فوق، پورت های ۱ تا ۲۰ از آدرس آیپی 89.43.3.170 بررسی شده اند و همان طور که در تصویر قابل مشاهده است، پورت های باز و غیر بسته به همراه پروتکلی که در این پورت ها مورد استفاده قرار گرفته است قابل مشاهده است.

در ادامه تصاویری از اجرای دستوراتی که در صورت تمرین گفته شده بود به کمک نرم افزار nmap اجرا شوند آورده شده است:

TCP Full Scan

اسکن تمام پورت های TCP یک هاست و مشخص کردن وضعیت آن ها به کمک دستور Nmap -T4 -sT قابل انجام است و در ادامه تصویری از خروجی اجرای این دستور برای آدرس آیپی 89.43.3.66 در لینوکس آورده شده است:

```
h-user@h-primary:~$ sudo nmap -T4 -sT 89.43.3.66
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-15 22:10 +0330
Nmap scan report for 66.mobinn.net (89.43.3.66)
Host is up (0.059s latency).
Not shown: 991 closed ports
PORT      STATE      SERVICE
80/tcp    filtered  http
800/tcp   filtered  mdbs_daemon
801/tcp   filtered  device
1723/tcp  open      pptp
2000/tcp  open      cisco-sccp
8080/tcp  open      http-proxy
8291/tcp  filtered  unknown
8443/tcp  open      https-alt
8800/tcp  open      sunwebadmin

Nmap done: 1 IP address (1 host up) scanned in 6.31 seconds
h-user@h-primary:~$
```

همان طور که در تصویر بالا قابل مشاهده است، ۹۹۱ پورت این هاست بسته است و وضعیت سایر پورت ها به همراه شماره ی پورت در تصویر فوق قابل مشاهده است. به کمک این دستور می توان تمام پورت های TCP یک هاست را مورد بررسی قرار داد.

Stealth Scan

برای انجام این نوع اسکن از دستوری که در تصویر زیر قابل مشاهده است در لینوکس استفاده می شود. در این تصویر خروجی ای از اجرای این دستور در لینوکس نیز آورده شده است.

```
h-user@h-primary:~$ sudo nmap -sS 89.43.3.66
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-15 22:21 +0330
Nmap scan report for 66.mobinnet.net (89.43.3.66)
Host is up (0.062s latency).
Not shown: 991 closed ports
PORT      STATE      SERVICE
80/tcp    filtered  http
800/tcp   filtered  mdbus_daemon
801/tcp   filtered  device
1723/tcp  open      pptp
2000/tcp  open      cisco-sccp
8080/tcp  open      http-proxy
8291/tcp  filtered  unknown
8443/tcp  open      https-alt
8800/tcp  open      sunwebadmin

Nmap done: 1 IP address (1 host up) scanned in 4.59 seconds
```

این نوع اسکن، این قابلیت را برای هکر ها ایجاد می کند که بدون آن که آدرس آیپی آن ها فاش شود، یک هاست را مورد بررسی قرار دهند. دلیل آنکه آدرس آیپی هکر در چنین اسکنی فاش نمی شود نیز نوع connection ای است که در این ارتباط تشکیل داده می شود.

UDP Scan

برای انجام این نوع اسکن از دستوری که در تصویر زیر قابل مشاهده است در لینوکس استفاده می شود. در این تصویر خروجی ای از اجرای این دستور در لینوکس نیز آورده شده است.

```
h-user@h-primary:~$ sudo nmap -p 140-145 -sU 89.43.3.170
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-15 22:26 +0330
Nmap scan report for 170.mobinn.net (89.43.3.170)
Host is up (0.46s latency).

PORT      STATE SERVICE
140/udp    closed emfis-data
141/udp    closed emfis-ctrl
142/udp    closed bl-idm
143/udp    closed imap
144/udp    closed news
145/udp    closed uaac

Nmap done: 1 IP address (1 host up) scanned in 1.41 seconds
```

همان طور که در تصویر بالا قابل مشاهده است، به کمک زبانه ی -sU می توان پورت های UDP را بررسی کرد و به کمک زبانه ی -p نیز می توان محدوده ی پورت های مورد نظر را بررسی کرد.

Fingerprint Scan

با توجه به جستجویی که درباره ی fingerprint در شبکه انجام شد، این ابزار در اصل این قابلیت را ایجاد می کند که به کمک آن می توان اطلاعاتی را درباره ی سیستم عاملی که بر روی هاست های مختلف و software application هایی که بر روی هاست های مختلف قرار دارند را کسب کرد.

برای انجام این نوع اسکن از دستوری که در تصویر زیر قابل مشاهده است در لینوکس استفاده می شود. در این تصویر خروجی ای از اجرای این دستور در لینوکس نیز آورده شده است.

```
vnigh-primary:~$ sudo nmap 89.43.3.170 -O
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-15 23:25 +0330
Nmap scan report for 170.mobinn.net (89.43.3.170)
Host is up (0.40s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    filtered smtp
80/tcp    open  http
443/tcp   open  https
1723/tcp  open  pptp
2000/tcp  open  cisco-sccp
6881/tcp  filtered bittorrent-tracker
6901/tcp  filtered jetstream
6969/tcp  filtered acnsoda
8291/tcp  filtered unknown
Aggressive OS guesses: Grandstream GXV3275 video phone (94%), Linux 3.2 - 3.8 (94%), Linux 3.3 (93%), Linux 2.6.32 - 2.6.39 (92%), Linux 2.6.32 (90%), Linux 2.6.32 or 3.10 (90%), Linux 3.4 (90%), Linux 3.8 (90%), WatchGuard Firewall 11.8 (90%), Linux 3.11 - 4.1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 24 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 171.11 seconds
```

همان طور که در انتهای خروجی داده شده در این تصویر مشخص است، این دستور اطلاعاتی را درباره ی سیستم عامل های مورد استفاده در اختیارمان می گذارد.

Idle Scan

برای انجام این نوع اسکن از دستوری که در تصویر زیر قابل مشاهده است در لینوکس استفاده می شود. در این تصویر خروجی ای از اجرای این دستور در لینوکس نیز آورده شده است و در ادامه توضیحاتی داده خواهد شد.

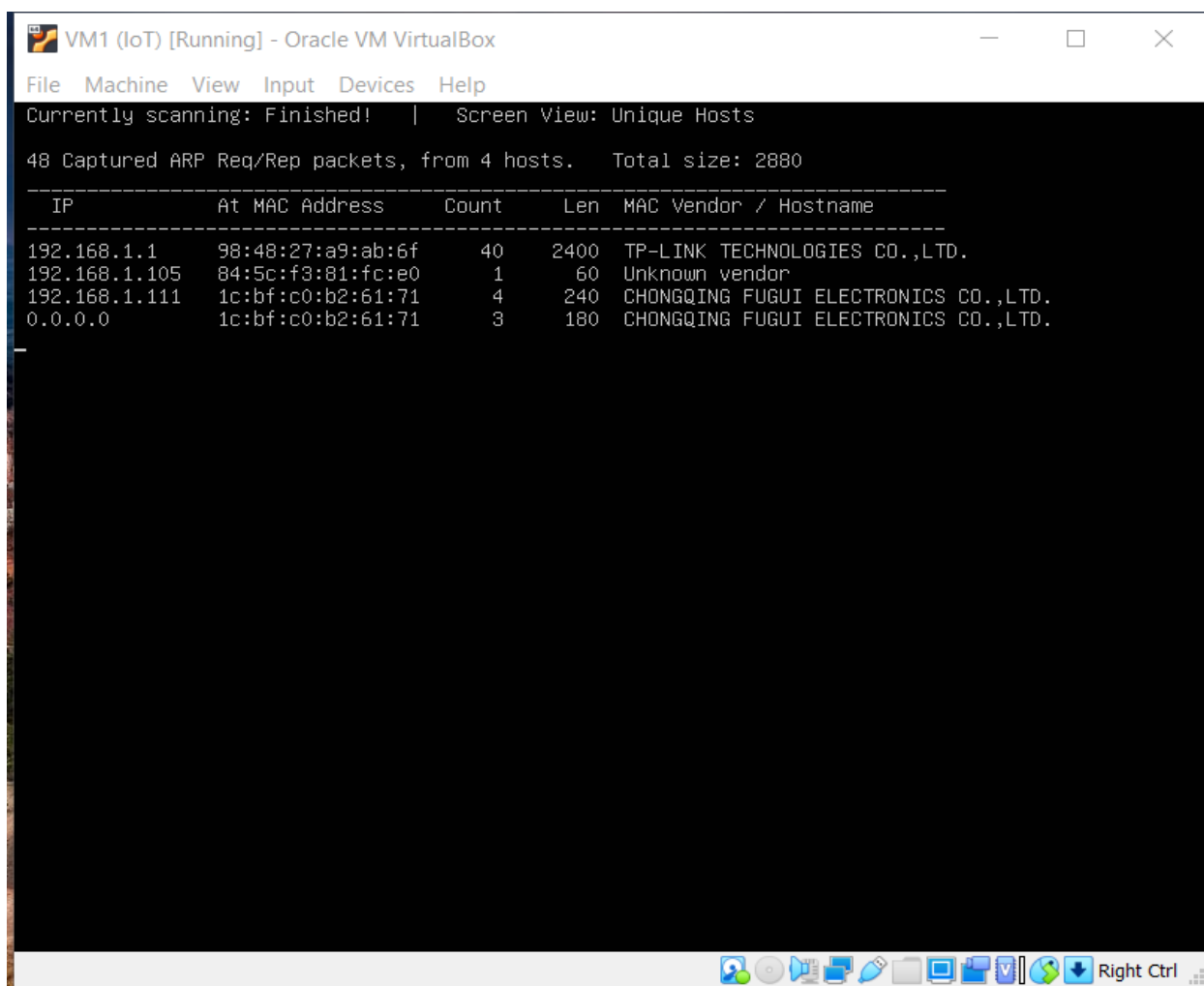
```
vm1gh-primary:/$ sudo nmap -Pn -sI 192.168.1.108 -p80 89.43.3.170
[sudo] password for vm1:
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-16 18:07 +0330
Idle scan zombie 192.168.1.108 (192.168.1.108) port 80 cannot be used because IP ID sequence class is: All zeros. Try another proxy.
QUITTING!
vm1gh-primary:/$
```

این نوع اسکن در اصل این قابلیت را ایجاد می کند که هکر بدون آنکه آدرس آیپی اش فاش شود، یک هاست و پورت های آن را مورد بررسی قرار دهد و در این نوع اسکن، آدرس آیپی ای که برای اسکن پورت های هاست مورد نظر مورد استفاده قرار می گیرد، آدرس یک هاست زامبی است و نه هاست اصلی هکر.

همان طور که در تصویر بالا قابل مشاهده است، به دلیل آنکه من آدرس آیپی یک سیستم زامبی را در اختیار نداشتم که به عنوان ورودی به این دستور بدهم و آدرس آیپی سیستم خودم را دادم، این دستور به شکل کامل انجام نشد و نتوانستم پورت های آدرس آیپی 89.43.3.170 را بررسی کنم.

ابزار netdiscover:

ابزاری است که عمدتاً برای بررسی هاست های فعال در یک شبکه مورد استفاده قرار می گیرد و به کمک آن می توان هاست های فعال شبکه ی داخلی ای را که از طریق آن به اینترنت متصل شده ایم را به دست آورد. برای استفاده از این ابزار، ubuntu server ای بر روی یک ماشین مجازی نصب شد و سپس همان طور که در تصویر زیر قابل مشاهده است، به کمک دستور `sudo netdiscover -r 89.43.3.0/16` قصد داشتم که آدرس آیپی فعال موجود در این محدوده را بررسی کنم اما با توجه به کاربردی که این نرم افزار دارد، صرفاً هاست های فعالی که در شبکه ی داخلی ای که به آن متصل بودم را در اختیارم قرار داد:



```
VM1 (IoT) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Currently scanning: Finished! | Screen View: Unique Hosts
48 Captured ARP Req/Rep packets, from 4 hosts. Total size: 2880
-----
IP                At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.1.1       98:48:27:a9:ab:6f  40     2400 TP-LINK TECHNOLOGIES CO.,LTD.
192.168.1.105     84:5c:f3:81:fc:e0   1        60 Unknown vendor
192.168.1.111     1c:bf:c0:b2:61:71   4       240 CHONGQING FUGUI ELECTRONICS CO.,LTD.
0.0.0.0           1c:bf:c0:b2:61:71   3       180 CHONGQING FUGUI ELECTRONICS CO.,LTD.
```

همان طور که در تصویر فوق مشخص است، ۴ هاست در شبکه ی داخلی فعال هستند که آدرس آیپی آن ها توسط نرم افزار netdiscover در لینوکس قابل مشاهده است.

ابزار hping3:

این ابزار ابزاری است که از آن عمدتاً برای ping گرفتن از آدرس های آیپی در سیستم عامل لینوکس استفاده می شود. در قسمت اول از بخش اول، آدرس آیپی ای که ping گرفته شد آدرس آیپی مربوط به www.google.com بود و این آدرس آیپی عبارت است از 216.58.209.132 و در این قسمت برای بررسی صحت ping ای که در بخش اول گرفته شد، همین آدرس آیپی به کمک hping3 در لینوکس مورد بررسی قرار گرفت و تصویر خروجی آن در ادامه آورده شده است:

```
h-user@h-primary:~$ sudo hping3 -S 216.58.209.132 -p 80 -c 10
HPING 216.58.209.132 (tun1 216.58.209.132): S set, 40 headers + 0 data bytes
len=44 ip=216.58.209.132 ttl=60 id=16244 sport=80 flags=SA seq=0 win=65535 rtt=451.8 ms
len=44 ip=216.58.209.132 ttl=60 id=28985 sport=80 flags=SA seq=1 win=65535 rtt=451.0 ms
len=44 ip=216.58.209.132 ttl=60 id=29676 sport=80 flags=SA seq=2 win=65535 rtt=498.3 ms
len=44 ip=216.58.209.132 ttl=60 id=26409 sport=80 flags=SA seq=3 win=65535 rtt=476.4 ms
len=44 ip=216.58.209.132 ttl=60 id=27901 sport=80 flags=SA seq=5 win=65535 rtt=442.6 ms
len=44 ip=216.58.209.132 ttl=60 id=39858 sport=80 flags=SA seq=4 win=65535 rtt=1472.7 ms
len=44 ip=216.58.209.132 ttl=60 id=28804 sport=80 flags=SA seq=6 win=65535 rtt=451.1 ms
len=44 ip=216.58.209.132 ttl=60 id=29666 sport=80 flags=SA seq=7 win=65535 rtt=453.6 ms
len=44 ip=216.58.209.132 ttl=60 id=7908 sport=80 flags=SA seq=8 win=65535 rtt=460.3 ms
len=44 ip=216.58.209.132 ttl=60 id=34252 sport=80 flags=SA seq=9 win=65535 rtt=447.5 ms

--- 216.58.209.132 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 442.6/560.5/1472.7 ms
```

همان طور که در تصویر فوق قابل مشاهده است، از زبانه ی C- برای گرفتن پینگ استفاده شده است و عددی که پس از این زبانه می آید، مشخص کننده ی تعداد بسته هایی است که کاربر قصد دارد برای پینگ گرفتن به هاست مورد نظر ارسال کند.

با توجه به این خروجی می توان نتیجه گرفت که هاست مورد نظر با آدرس 216.58.209.132 همان طور که در بخش اول مشخص شد، در دسترس است.

یافتن اطلاعات بیشتر درباره ی هاست های مورد نظر به کمک ابزار whatweb:

به کمک این ابزار می توان اطلاعاتی از قبیل سیستم عاملی که بر روی یک هاست قرار دارد، کشوری که هاست مورد نظر در آن قرار دارد، در دسترس بودن یا نبودن یک هاست و... به دست آورد.

برای بررسی این نرم افزار آدرس آیپی 89.43.3.170 که در در محدوده ی تعیین شده در تمرین قرار دارد مورد استفاده قرار گرفت و نتیجه ی خروجی آن در تصویر زیر قابل مشاهده است.

```
vm10h-primary:/$ whatweb 89.43.3.170
/usr/lib/ruby/vendor_ruby/target.rb:188: warning: URI.escape is obsolete
http://89.43.3.170 [200 OK] Country[ROMANIA][RO], IP[89.43.3.170], Mikrotik-RouterOS[6.48.3][Telnet], PasswordField, Script, Title[RouterOS route
configuration page]
```

همان طور که در تصویر بالا قابل مشاهده است، هاست 89.43.3.170 در کشور رمانی قرار دارد و در دسترس است زیرا کد ۲۰۰ توسط این هاست باز گردانده شده است و همچنین Router OS ای که در این نرم افزار مورد استفاده قرار گرفته است، Telnet است و ورژن آن نیز 6.48.3 می باشد.