

پروژه اول درس امنیت اطلاعات

نام دانشجو: رهام زنده دل نوبری

شماره دانشجویی: 9731088

1. بخش اول

گرفتن Ping از آی پی مشخص

کد مربوطه: ping.py

برای این قسمت یک تابع جداگانه نوشته شده است که ابتدا از کاربر خواسته می‌شود تا IP یا domain مورد نظر خود را وارد کند. بعد از وارد کردن، در صورتی که domain باشد به کمک کتابخانه‌ی socket، IP آدرس مربوطه را بدست می‌آوریم و IP و domain را برمی‌گردانیم.

```
def get_host_name_ip():
    get_input = input("Enter your ip address/domain: ")
    try:
        if re.search("[a-zA-Z]", get_input):
            ip_address = socket.gethostbyname(get_input)
            domain_name = get_input
        else:
            ip_address = get_input
            domain_name = ""
    except:
        print("Unable to get Hostname and IP")
    return domain_name, ip_address
```

برای بدست آوردن domain نیز می‌توان از دستور socket.gethostbyaddr() استفاده کرد.

بعد از بدست آوردن آدرس IP، تابع دیگری صدا زده می‌شود که در آن در صورت وجود نداشتن فایل txt با نام مناسب ایجاد می‌شود و متن‌های مناسب چاپ می‌شود.

```
def create_file():
    try:
        f = open("result_ping.txt", "x")
        f.write("Ping a range of IPs and find active hosts:\n ")
        print("New file created!")
    except:
        print("File already exists!")
```

در نهایت با کمک کتابخانه‌ی os، آدرس IP مورد نظر ping می‌شود و نتایج چاپ می‌شود.

```
domain, ip = get_host_name_ip()
create_file()
```

```
print("Pinging " + domain + " : " + ip)
response = os.system("ping " + ip)
file = open("result_ping.txt", "a")
file.write("\n")
file.write(os.popen(f"ping {ip}").read())
file.write("\n")
file.write("-----")
```

دو روش مختلف `os.system("ping " + ip)` و `os.popen(f"ping {ip}")` برای گرفتن پینگ استفاده شده است.

```
PS D:\Uni\Information Security\Project 1> & C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe "d:\Uni\Information Security\Project 1\ping.py"
Enter your ip address/domain: google.com
New file created!
Pinging google.com : 216.58.214.14

Pinging 216.58.214.14 with 32 bytes of data:
Reply from 216.58.214.14: bytes=32 time=141ms TTL=59
Reply from 216.58.214.14: bytes=32 time=171ms TTL=59
Reply from 216.58.214.14: bytes=32 time=153ms TTL=59
Reply from 216.58.214.14: bytes=32 time=141ms TTL=59

Ping statistics for 216.58.214.14:
    Approximate round trip times in milli-seconds:
        Minimum = 141ms, Maximum = 171ms, Average = 151ms
```

شکل 1-1 نمونه ای از اجرای کد که در آن `google.com` را پینگ کرده ایم.

اسکن یک محدوده آیپی و یافتن هاست های فعال

کد مربوطه: `scanIPs.py`

برای این قسمت یک تابع جداگانه نوشته شده است که ابتدا از کاربر خواسته می شود تا IP آدرس مورد نظر خود را وارد کند. بعد از وارد کردن IP را با دستور `spilt` در جاهایی که "." وجود دارد جدا می کند و در نهایت از سومین "." به بعد کنار می گذارد و از تیکه ی اول استفاده می کند و `return` می کند. همچنین از کاربر خواسته می شود تا با گرفتن `start` و `end` نیز `range` داده را مشخص کند.

```
def get_host_name_ip():
    ip_address = input("Enter your network address: ")
    groups = ip_address.split('.')
    ip_address = '.'.join(groups[:3])
    start = input("Enter the starting number: ")
    end = input("Enter the last number: ")
    return ip_address, start, end
```

بعد از بدست آوردن آدرس IP تابع دیگری صدا زده می شود که در آن در صورت وجود نداشتن فایل `txt` با نام مناسب ایجاد می شود و متن های مناسب چاپ می شود.

```
def create_file():
    try:
        f = open("result_scanIPs.txt", "x")
        f.write("Ping a range of IPs and find active hosts.\n ")
```

```

        print("New file created!")
    except:
        print("File already exists!")

```

در نهایت تابع آخر صدا زده می‌شود که در آن با یک `for` و کتابخانه‌ی `os`، تمامی `host` های فعال و `live` پیدا می‌شوند و علاوه بر چاپ شدن در فایل مورد نظر نیز ذخیره می‌شوند.

```

def find_active_hosts(ip, start, end):
    file = open("result_scanIPs.txt", "a")
    file.write("\n")
    for i in range(end - start + 1):
        host = ip + "." + str(start + i)
        response = os.popen(f"ping {host}").read()
        if "Received = 4" in response:
            print(f"UP {host} Ping Successful ---> Live")
            file.write(" %s ---> Live " % host)
            file.write("\n")

```

با استفاده از `socket` نیز می‌توان `live` بودن `host` را چک کرد.

```

PS D:\Uni\Information Security\Project 1> & C:/Users/asus/AppData/Local/Programs/Python/Python39/python.exe "d:/Uni/Information Security/Project 1/ScanIPs.py"
Enter your network address: 89.43.3.0
Enter the starting number: 0
Enter the last number: 255
File already exists!
Scanning in progress...
UP 89.43.3.1 Ping Successful ---> Live
UP 89.43.3.2 Ping Successful ---> Live
UP 89.43.3.3 Ping Successful ---> Live
UP 89.43.3.5 Ping Successful ---> Live
UP 89.43.3.6 Ping Successful ---> Live
UP 89.43.3.7 Ping Successful ---> Live
UP 89.43.3.8 Ping Successful ---> Live
UP 89.43.3.9 Ping Successful ---> Live
UP 89.43.3.10 Ping Successful ---> Live
UP 89.43.3.11 Ping Successful ---> Live
UP 89.43.3.12 Ping Successful ---> Live
UP 89.43.3.13 Ping Successful ---> Live

```

شکل 2-1 نمونه از اجرا شدن برنامه و اسکن شدن IP های آدرس 89.43.3.0

اسکن پورت‌های باز یک هاست فعال

کد مربوطه: `openPorts.py`

برای این قسمت یک تابع جداگانه نوشته شده است که ابتدا از کاربر خواسته می‌شود تا IP آدرس مورد نظر خود را وارد کند. همچنین از کاربر خواسته می‌شود تا با گرفتن `start` و `end` نیز `range`، `port` های مورد نظرش را مشخص کند.

```

def get_host_name_ip():
    ip_address = input("Enter your host IP address: ")
    start = input("Enter the start port number: ")
    end = input("Enter the last port number: ")
    return ip_address, start, end

```

بعد از بدست آوردن آدرس IP تابع دیگری صدا زده می‌شود که در آن در صورت وجود نداشتن فایل `txt` با نام مناسب ایجاد می‌شود و متن‌های مناسب چاپ می‌شود.

```
def create_file():
    try:
        f = open("result_openPorts.txt", "x")
        f.write("Scan an IP and find open ports:\n ")
        print("New file created!")
    except:
        print("File already exists!")
```

در نهایت تابع آخر صدا زده می‌شود که در آن با یک `for` و کتابخانه‌ی `socket`، IP آدرس مورد نظر را با تمامی `port`هایی که داخل `range` داده هستند چک می‌شود و در صورت برقراری `connection` که عدد 0 برمی‌گرداند، نتیجه چاپ در فایل مورد نظر ذخیره می‌شود.

```
def find_open_ports(ip, start, end):
    print("IP --> " + ip)
    file = open("result_openPorts.txt", "a")
    file.write("\n")
    file.write("IP %s : " % ip)
    for port in range(start, end):
        s = socket()
        conn = s.connect_ex((ip, port))
        if conn == 0:
            print("Port %d: OPEN" % (port,))
            file.write("\n")
            file.write("Port %d: OPEN" % (port,))
    s.close()
```

```
PS D:\Uni\Information Security\Project 1> & C:/Users/asus/AppData/Local/Programs/Python/Python39/python.exe "d:/Uni/Information Security/Project 1/OpenPorts.py"
Enter your host IP address: 89.43.3.120
Enter the start port number: 1
Enter the last port number: 500
File already exists!
Scanning in progress...
IP --> 89.43.3.120
Port 22: OPEN
Port 80: OPEN
Port 443: OPEN
```

شکل 3-1 نمونه ای از اجرای برنامه که در آن `port`های باز آدرس 89.43.3.120 را از 1 تا 500 چک کرده‌ایم.

2. بخش دوم

Nmap

این ابزار دست استفاده کننده های خود را با روش های مختلفی که برای تحلیل شبکه استفاده می کند، باز گذاشته است. در این بخش سعی می کنیم خروجی هایی که برای آیدی 89.43.3.120 را تولید کرده بودیم با تنظیمات مختلف این ابزار بدست آوریم و برای هر مورد توضیح مختصری بیاوریم.

TCP full scan

این روش هنگامی استفاده می شود که raw packet privileges را نداریم. به این معنی که نمی توانیم در ساخت پکت هایی که ساخته می شود دخالت کنیم. همانطور که می دانیم برخی از دستورات network و hardware، دستورات privilege هستند و این به منظور امنیت بیشتر است. به همین دلیل برای scan کردن port ها لازم است برای هر درخواست یک TCP connection توسط فراخوانی سیستمی connect ایجاد شود و پس از handshake کامل این بسته ارسال شود که بسیار زمان برتر از روش های بدون واسطه دیگر است. برای این که از این روش استفاده کنیم از دستور زیر استفاده می کنیم:

```
nmap 89.43.3.120 -sT
```

```
C:\Users\asus>nmap -sT 89.43.3.120
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-08 18:16 Iran Standard Time
Nmap scan report for 120.mobinnet.net (89.43.3.120)
Host is up (0.065s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp  open  mysql
10000/tcp open  snet-sensor-mgmt
Nmap done: 1 IP address (1 host up) scanned in 50.78 seconds
```

Stealth scan

این روش، روشی معروف و بسیار سریع است زیرا برخلاف روش قبل نیازی به اتصال کامل TCP نیست و فقط یک پیام SYN فرستاده می شود. این روش به raw packet privileges نیاز دارد زیرا دخالت در ساخت بسته ها و اجرای پروتکل دارد. این روش به قدری سریع است که می تواند چندین port را در ثانیه تحلیل کند. برای اجرای scan به کمک این روش، دستور زیر باید اجرا شود:

```
nmap 89.43.3.120 -sS
```

```
C:\Users\asus>nmap 89.43.3.120 -sS
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-08 19:44 Iran Standard Time
Nmap scan report for 120.mobinn.net (89.43.3.120)
Host is up (0.075s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp  open  mysql
8291/tcp  filtered unknown
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 10.94 seconds
```

UDP scan

این روش برای پیدا کردن port های پروتکل UDP استفاده می شود. از آنجایی که پیدا کردن این پورت ها با این پروتکل زمان گیر است معمولاً به آنها اهمیتی داده نمی شود. اما بسیاری از پروتکل های معروف بر روی این پروتکل ساخته شده اند همانند DNS, DHCP, SNMP. برای استفاده از پروتکل، دستور زیر را اجرا می کنیم:

```
nmap 89.43.3.120 -sU
```

```
C:\Users\asus>nmap 89.43.3.120 -sU
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-08 19:51 Iran Standard Time
Nmap scan report for 120.mobinn.net (89.43.3.120)
Host is up (0.060s latency).
Not shown: 993 closed udp ports (port-unreach)
PORT      STATE SERVICE
67/udp    open|filtered dhcp
68/udp    open|filtered dhcp
137/udp    open|filtered netbios-ns
138/udp    open|filtered netbios-dgm
139/udp    open|filtered netbios-ssn
1900/udp   open|filtered upnp
10000/udp  open  ndmp

Nmap done: 1 IP address (1 host up) scanned in 1102.50 seconds
```

Fingerprint scan

از این scan برای دستیابی یا حدس زدن OS مربوط به مقصد استفاده می شود. برای استفاده از این روش، دستور زیر را اجرا می کنیم:

```
nmap 89.43.3.120 -O
```

```

C:\Users\asus>nmap 89.43.3.120 -O
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-08 20:36 Iran Standard Time
Nmap scan report for 120.mobinn.net (89.43.3.120)
Host is up (0.081s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
443/tcp   open      https
3306/tcp   open      mysql
8291/tcp   filtered  unknown
10000/tcp  open      snet-sensor-mgmt
Aggressive OS guesses: Linux 2.6.32 (91%), Linux 2.6.32 or 3.10 (91%), Linux 3.5 (91%), Linux 4.2 (91%), Linux 4.4 (91%),
, Synology DiskStation Manager 5.1 (91%), WatchGuard Firewall 11.8 (91%), Linux 2.6.35 (90%), Linux 3.10 (90%), Linux 2.
6.39 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 12 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.69 seconds

```

Idle scan

در این روش از یک واسطه که **zombie** نامیده میشود برای تحلیل شبکه استفاده می‌شود. این روش به مخفی ماندن منبع تحلیل کننده کمک می‌کند. این **zombie** معمولاً یک سرور ویندوز قدیمی یا پرینتر قدیمی است زیرا سیستم‌های کنونی بدلیل داشتن **firewall** در مقابل **zombie** شدن ایمن هستند. **Nmap** یک **script** مخصوص دارد که به کمک آن کاندیدهای محتمل برای **zombie** شدن را به دست می‌آورد. هر چند تعداد زیادی از آن‌ها ایمن هستند.

```
nmap -p443 --script ipidseq -iR 1000 >> ipidseq_result.txt
```

نتیجه اجرای دستور فوق در فایل **ipidseq_result** قابل مشاهده است. ما قادر به پیدا کردن سیستم **zombie** نشدیم. با پیدا شدن **zombie** دستور زیر را اجرا می‌کنیم:

```
nmap 89.43.3.120 -sI <zombie IP/domain>
```

IP Scan

به کمک ابزار **Nmap** نیز می‌توان **range**، آدرس IP های فعال را بدست آورد. دستور آن به صورت زیر است:

```
nmap -sn 89.43.3.0/20 >> result_ip_scan_nmap.txt
```

نتیجه این دستور در فایل **result_ip_scan_nmap** ذخیره شده است.

Whatweb

از این ابزار برای بررسی وبسایت‌ها استفاده می‌شود. اطلاعات مفیدی مانند تکنولوژی‌های استفاده شده برای قسمت‌های مختلف

یک وبسایت همانند **frontend, backend, webserver, os** و ... را به ما می‌دهد. دستور زیر لاگ **whatweb** برای آدرس

aut.ac.ir را در فایل **whatweb_log.txt** ذخیره می‌کند:

```
whatweb aut.ac.ir -v -log-verbose=whatweb_log.txt
```

Netdiscover

این ابزار برای بررسی شبکه‌های داخلی و دستگاه‌های متصل به شبکه داخلی است و به دردمان نمی‌خورد.

Hping3

ابزار دیگری است که به کمک آن می‌توان تمرین اول یعنی IP ping را تست کرد. نتیجه‌ی آن به صورت زیر قابل مشاهده است:

```
rohamzn@ubuntu:~$ sudo hping3 --tracerout -V -1 aut.ac.ir
using ens33, addr: 192.168.232.128, MTU: 1500
HPING aut.ac.ir (ens33 185.211.88.131): icmp mode set, 28 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=192.168.232.2 name=_gateway
hop=1 hoprtt=11.9 ms
hop=2 TTL 0 during transit from ip=192.168.1.254 name=UNKNOWN
hop=2 hoprtt=12.0 ms
hop=3 TTL 0 during transit from ip=89.219.192.1 name=UNKNOWN
hop=3 hoprtt=36.0 ms
hop=4 TTL 0 during transit from ip=10.22.26.102 name=UNKNOWN
hop=4 hoprtt=35.8 ms
hop=5 TTL 0 during transit from ip=10.22.26.101 name=UNKNOWN
hop=5 hoprtt=32.2 ms
hop=6 TTL 0 during transit from ip=10.201.203.30 name=UNKNOWN
hop=6 hoprtt=36.0 ms
^C
--- aut.ac.ir hping statistic ---
7 packets transmitted, 6 packets received, 15% packet loss
round-trip min/avg/max = 11.9/27.3/36.0 ms
```

Xprobe2

این ابزار به منظور شناسایی یا حداقل حدس سیستم عامل هدف با بررسی‌های موازی‌ای که انجام می‌دهد و مقایسه‌ی نتایج با database خود طراحی شده است. دستور زیر را اجرا می‌کنیم:

```
sudo xprobe2 -AT1700-2000 89.43.3.120
```

از xprobe2 برای چک کردن پورت‌های باز نیز استفاده می‌شود. نتایج این دستور در فایل xprobe_port_scan.txt موجود است.


اطلاعات دیگر


از سایت www.ip2location.com استفاده می‌کنیم برای گرفتن اطلاعات گوناگون در مورد هاست مورد نظر که در عکس‌های زیر آنرا مشاهده می‌کنیم.

IP Address 89.43.3.120 Demo

We offer free IP geolocation query up to 50 IP addresses per day. [Sign up](#) for a demo account to be entitled to a higher daily limit. You still have **49/50** query limit available for [today](#).




89.43.3.120

 LOOKUP

 This demo uses data from **IP2Location DB25** geolocation database and **IP2Proxy PX11** anonymous proxy database for results.

IP Lookup Result

[Share The Result](#)

Permalink	https://www.ip2location.com/89.43.3.120 
<input checked="" type="checkbox"/> IP Address	89.43.3.120
<input checked="" type="checkbox"/> Country	 Iran (Islamic Republic of) [IR] 
<input type="checkbox"/> Region	Hormozgan
<input type="checkbox"/> City	Kish
<input type="checkbox"/> Coordinates of City[†]	26.557780, 54.019440 (26°33'28"N 54°1'10"E)
<input type="checkbox"/> ISP	Mobin Net Communication Company (Private Joint Stock)
<input type="checkbox"/> Local Time	09 Nov, 2022 12:18 AM (UTC +03:30)
<input type="checkbox"/> Domain	mobinnet.net
<input type="checkbox"/> Net Speed	(DSL) Broadband/Cable/Fiber/Mobile
<input type="checkbox"/> IDD & Area Code	(98) 076
<input type="checkbox"/> ZIP Code	-

Bots

You can easily lookup an IP address on the below channels using the below commands.

Twitter Bot

IP2Location Twitter Bot	@ip2location 89.43.3.120
IP2Proxy Twitter Bot	@ip2proxybot 89.43.3.120

Slack Bot

IP2Location Slack Bot	/ip2location 89.43.3.120
IP2Proxy Slack Bot	/ip2proxy 89.43.3.120

Reddit Bot

IP2Location Reddit Bot	u/ip2location_bot 89.43.3.120
IP2Proxy Reddit Bot	u/ip2proxy_bot 89.43.3.120

Telegram Bot

IP2Location Telegram Bot	ip2location 89.43.3.120
IP2Proxy Telegram Bot	ip2proxy 89.43.3.120