تمرین عملی دوم - صبا سحبان ۹۹۳۱۰۹۶

توضیح کد:

این کد پایتون یک ابزار ساده اسکن شبکه است که به ما این امکان را می‌دهد که یا یک محدوده آدرس IP یا پورت‌های خاص را در یک محدوده از آدرس‌های IP اسکن کنیم و یک سری اطلاعاتی از قبیل پورت‌ها و آی‌پی‌های اکتیو را دریافت کنیم.

Is_host_alive:

```
def is_host_alive(ip):
    try:
        response = os.system(f"ping -c 3 -W 1 {ip}")
        return response == 0
    except Exception as e:
        print(f"Error while pinging {ip}: {e}")
        return False
```

- این تابع از os.system برای اجرای دستور ping به منظور بررسی اینکه آیا یک میزبان زنده است یا خیر استفاده می‌کند. اگر میزبان قابل دسترس و به ping پاسخ دهد، True برمی‌گرداند؛ در غیر این صورت False.

Scan_ip_range:

```
def scan_ip_range(start_ip, end_ip):
    active_ips = []
    for ip in range(int(start_ip.split('.')[-1]), int(end_ip.split('.')[-1]) +
1):
        current_ip = ".".join(start_ip.split('.')[:-1]) + '.' + str(ip)
        if is_host_alive(current_ip):
            active_ips.append(current_ip)
    return active_ips
```

- یک محدوده از آدرس‌های IP بین start_ip و end_ip را اسکن می‌کند.
- از تابع is_host_alive برای شناسایی آدرس‌های IP فعال استفاده می‌کند.
- یک لیست از آدرس‌های IP فعال را برمی‌گرداند.

Scan_tcp_ports:

```
def scan_tcp_ports(ip, start_port, end_port):
    open_tcp_ports = []
    for port in range(start_port, end_port + 1):
        if is_port_open(ip, port, socket.SOCK_STREAM):
            open_tcp_ports.append(port)
    return open_tcp_ports
```

- پورت‌های باز TCP را در یک آدرس IP خاص (ip) داخل محدوده پورت داده شده اسکن می‌کند.
- از تابع is_port_open برای بررسی اینکه آیا پورت باز است یا خیر استفاده می‌کند.
- یک لیست از پورت‌های باز TCP را برمی‌گرداند.

Scan_udp_ports:

```
def scan_udp_ports(ip, start_port, end_port):
    open_udp_ports = []
    for port in range(start_port, end_port + 1):
        if is_port_open(ip, port, socket.SOCK_DGRAM):
            open_udp_ports.append(port)
    return open_udp_ports
```

- پورت‌های باز UDP را در یک آدرس IP خاص (ip) داخل محدوده پورت داده شده اسکن می‌کند.
- از تابع is_port_open برای بررسی اینکه آیا پورت باز است یا خیر استفاده می‌کند.
- یک لیست از پورت‌های باز UDP را برمی‌گرداند.

Is_port_open / is_port_open_udp

```
def is_port_open(ip, port, sock_type):
    try:
        with socket.socket(socket.AF_INET, sock_type) as s:
            s.settimeout(1)
            s.connect((ip, port))
        return True
    except (socket.timeout, socket.error):
        return False
def is_port_open_udp(target, port):
    try:
        sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        sock.settimeout(1)
        result = sock.connect_ex((target, port))
        if result == 0:
            return True

    except (socket.timeout, socket.error):
        return False
```

- بررسی می‌کند آیا یک پورت خاص در یک آدرس IP داده شده با یک نوع سوکت مشخص (TCP یا UDP) باز است یا خیر.

Generate_report:

```
def generate_report(ip_range, active_ips, open_tcp_ports, open_udp_ports):
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    report_content = f"Scan Report - {timestamp}\n"
    report_content += f"IP Range: {ip_range}\n\n"

    if active_ips:
```

```
        report_content += "Active IP Addresses:\n"
        for ip in active_ips:
            report_content += f"- {ip}\n"
        report_content += "\n"

    if open_tcp_ports:
        report_content += "Open TCP Ports:\n"
        for port in open_tcp_ports:
            report_content += f"- Port {port} is open\n"
        report_content += "\n"

    if open_udp_ports:
        report_content += "Open UDP Ports:\n"
        for port in open_udp_ports:
            report_content += f"- Port {port} is open\n"
        report_content += "\n"

    return report_content
```

- یک گزارش شامل اطلاعات در مورد محدوده IP اسکن شده، آدرس‌های IP فعال و پورت‌های باز TCP/UDP را تولید می‌کند.
- محتوای گزارش را به صورت یک رشته برمی‌گرداند.

Save_report:

```
def save_report(filename, content):
    with open(filename, 'a') as file:
        file.write(content)
```

- تابع save_report در نهایت محتوای گزارش را در فایل append میکند.

بخش ۱)

۱-۱-۱- Scanning در زبان تست نفوذ به اسکن یا بررسی شبکه و سیستم‌ها به منظور جمع‌آوری اطلاعات پرداخته می‌شود. این فعالیت معمولاً به منظور شناسایی سرویس‌ها، پورت‌ها، و ممکن است به تحلیل نقاط ضعف سیستم هدف نیز منجر شود.

- در فاز Scanning، حاصل اطلاعات می‌تواند شامل موارد زیر باشد:
- پورت‌های باز: شناسایی پورت‌های باز در سیستم یا شبکه هدف.
- آدرس‌های IP فعال: تشخیص ماشین‌هایی که در شبکه فعال هستند.
- سرویس‌ها: شناسایی نوع سرویس‌های در حال اجرا بر روی پورت‌های باز.

۱-۱-۲- Footprinting:

- Footprinting یا جمع‌آوری اطلاعات اولیه، مرحله‌ای است که قبل از Scanning انجام می‌شود.

- در این مرحله، اطلاعات جمع‌آوری می‌شوند که می‌توانند شامل اطلاعات DNS، WHOIS، شبکه‌های اجتماعی، و غیره باشند.
- هدف Footprinting، یافتن و تجمیع اطلاعات عمومی و اولیه درباره هدف است.

Scanning:

- Scanning مرحله‌ای بعد از Footprinting است که بیشتر متمرکز بر بررسی فعالیت شبکه و سیستم‌هاست.
- در این مرحله، به سرویس‌ها، پورت‌ها، و دستگاه‌های فعال در شبکه توجه می‌شود.
- هدف Scanning، بررسی و تشخیص نقاط قوت و ضعف امنیتی در سیستم یا شبکه هدف است.

۱-۱-۳-

راه‌های مقابله با Scanning:

1. استفاده از فایروال:
- تنظیم فایروال به نحوی که اسکن‌های ناخواسته را محدود کند یا متوقف کند.

2. شناسایی و اصلاح نقاط ضعف:
- اجرای تست‌های نفوذ داخلی به منظور شناسایی نقاط ضعف و اصلاح آن‌ها.

3. مدیریت دقیق پورت‌ها:
- تنظیم مدیریت دقیق پورت‌ها به منظور کاهش پوشش سطح حمله.

4. شناسایی و اصلاح اطلاعات WHOIS و DNS:
- مدیریت صحیح اطلاعات WHOIS و DNS به منظور کاهش اطلاعاتی که در دسترس هکر قرار می‌گیرد.

بررسی خروجی کد:

Ipscan:

قسمتی از خروجی

```
sabasahban@sabas-MacBook-Pro pythonProject5 % sudo python Scanner.py --ipscan -m 24 -ip 89.43.3.0 89.43.3.255
Active IP addresses: ['89.43.3.0', '89.43.3.1', '89.43.3.2', '89.43.3.3', '89.43.3.4', '89.43.3.5', '89.43.3.6', '89.43.3.7', '89.43.3.8', '89.43.3.9', '89.43.3.10', '89.43.3.11', '89.43.3.12', '89.43.3.13', '89.43.3.14', '89.43.3.15', '89.43.3.16', '89.43.3.17', '89.43.3.18', '89.43.3.19', '89.43.3.20', '89.43.3.21', '89.43.3.22', '89.43.3.23', '89.43.3.24', '89.43.3.25', '89.43.3.26', '89.43.3.27', '89.43.3.28', '89.43.3.29', '89.43.3.30', '89.43.3.31', '89.43.3.32', '89.43.3.33', '89.43.3.34', '89.43.3.35', '89.43.3.36', '89.43.3.37', '89.43.3.38', '89.43.3.39', '89.43.3.40', '89.43.3.41', '89.43.3.42', '89.43.3.43', '89.43.3.44', '89.43.3
```

TCP portscan:

```
sabasahban@sabas-MacBook-Pro pythonProject5 %   sudo python Scanner.py --portscan --tcp 1 7000 -ip 127.0.0.1 127.0.0.1

Open TCP ports on 127.0.0.1: [1082, 3306, 5000, 5431, 7000]
```

UDP portscan:

قسمتی از خروجی:

```
sabasahban@sabas-MacBook-Pro pythonProject5 % sudo python Scanner.py --portscan --udp 137 138 -ip 127.0.0.1 127.0.0.1

Open UDP ports on 127.0.0.1: [137, 138]
```

بخش دوم:

۲-۱-۱-

1. sS:

- این سوییچ اسکن SYN را نشان می‌دهد. در واقع، این اسکن SYN را انجام داده و در صورت دریافت یک پاسخ SYN/ACK، پورت به عنوان باز تلقی می‌شود.

2. sV:

- این سوییچ اطلاعات نسخه سرویس‌ها را نمایش می‌دهد. با استفاده از این سوییچ، می‌توان اطلاعات دقیق‌تری از نسخه سرویس‌های در حال اجرا در سیستم هدف کسب کرد.

3. sT:

- این سوییچ اسکن CONNECT را نشان می‌دهد. در این نوع اسکن، nmap یک اتصال TCP کامل برقرار می‌کند تا مشخص شود آیا پورت باز است یا خیر.

۲-۱-۲-

1. F:

- کاربرد: این سوییچ باعث می‌شود nmap فقط بر روی 100 پورت محبوب اسکن اجرا شود.
- تفاوت: این سوییچ از تعداد پورت‌های کمتری برای اسکن استفاده می‌کند (100 پورت محبوب)، که منجر به اجرای سریع‌تر و کم‌مخاطره‌تر می‌شود. این برخلاف اسکن استاندارد nmap است که تعداد بسیار بیشتری از پورت‌ها را اسکن می‌کند.

2. O:

- کاربرد: این سوییچ برای تشخیص نوع سیستم عامل (OS) در سرور مقصد استفاده می‌شود.
- تفاوت: این سوییچ با استفاده از پاسخ‌های ICMP و تجزیه و تحلیل fingerprintهای TCP، سعی می‌کند نوع سیستم عامل را تشخیص دهد. این کمک می‌کند تا در تست نفوذ، نوع سیستم عامل مورد هدف برای حملات متناسب انتخاب شود.

3. A:

- کاربرد: این سوییچ معانی مختلفی دارد: شامل اسکن‌های سریع TCP، اسکن‌های UDP، اسکن‌های OS detection، و اطلاعات ویژه بیشتر. این سوییچ برای تست نفوذ جامع‌تر به کار می‌رود.

- تفاوت: این سوییچ شامل یک مجموعه گسترده‌تر از اطلاعات و امکانات است. به عنوان مثال، علاوه بر اسکن TCP و UDP، شامل اطلاعات OS detection و اطلاعات جزئیاتی از سرویس‌های در حال اجرا می‌شود. این سوییچ برای تست‌های نفوذ جامع‌تر و جزئی‌تر استفاده می‌شود.

۲-۱-۳-

:sn .7

-این سوییچ یک اسکن ping را نمایش می‌دهد. یعنی تنها ICMP echo request می‌فرستد تا وضعیت زنده‌بودن دستگاه‌ها را بررسی کند.

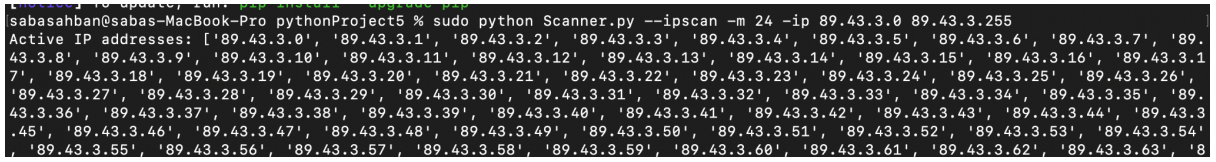
:pn.8

- این سوییچ باعث می‌شود nmap از اسکن ping صرف‌نظر کند و مستقیماً به اسکن پورت بپردازد. معمولاً برای مواردی که اسکن ping موفق نیست ولی هنوز می‌خواهیم اسکن پورت انجام شود، مورد استفاده قرار می‌گیرد.

مقایسه نتایج nmap و کد:

lpscan:

```
sabasahban@sabas-MacBook-Pro pythonProject5 % nmap -sn 89.43.3.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-29 01:26 +0330
Nmap scan report for 89.43.3.0
Host is up (0.00066s latency).
Nmap scan report for 1.mobinnet.net (89.43.3.1)
Host is up (0.00025s latency).
Nmap scan report for 2.mobinnet.net (89.43.3.2)
Host is up (0.00024s latency).
Nmap scan report for 3.mobinnet.net (89.43.3.3)
Host is up (0.00021s latency).
Nmap scan report for 4.mobinnet.net (89.43.3.4)
Host is up (0.00015s latency).
Nmap scan report for 5.mobinnet.net (89.43.3.5)
Host is up (0.00012s latency).
Nmap scan report for 6.mobinnet.net (89.43.3.6)
Host is up (0.00071s latency).
Nmap scan report for 7.mobinnet.net (89.43.3.7)
Host is up (0.00066s latency).
Nmap scan report for 8.mobinnet.net (89.43.3.8)
Host is up (0.00032s latency).
Nmap scan report for 9.mobinnet.net (89.43.3.9)
Host is up (0.00040s latency).
Nmap scan report for 10.mobinnet.net (89.43.3.10)
Host is up (0.00058s latency).
Nmap scan report for 11.mobinnet.net (89.43.3.11)
Host is up (0.00032s latency).
Nmap scan report for 12.mobinnet.net (89.43.3.12)
Host is up (0.00029s latency).
Nmap scan report for mx2.payaco-mnp.com (89.43.3.13)
Host is up (0.00030s latency).
Nmap scan report for 14.mobinnet.net (89.43.3.14)
Host is up (0.00028s latency).
Nmap scan report for 15.mobinnet.net (89.43.3.15)
Host is up (0.00061s latency).
Nmap scan report for 16.mobinnet.net (89.43.3.16)
Host is up (0.00041s latency).
Nmap scan report for 17.mobinnet.net (89.43.3.17)
Host is up (0.00028s latency).
Nmap scan report for 18.mobinnet.net (89.43.3.18)
Host is up (0.00071s latency).
Nmap scan report for 19.mobinnet.net (89.43.3.19)
```



UDP portscan:

```
sabasahban@sabas-MacBook-Pro pythonProject5 % sudo python Scanner.py --portscan --udp 137 138 -ip 127.0.0.1 127.0.0.1

Open UDP ports on 127.0.0.1: [137, 138]
```

```
sabasahban@sabas-MacBook-Pro pythonProject5 %  sudo nmap -sU 127.0.0.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-29 02:50 +0330
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000054s latency).
Not shown: 996 closed udp ports (port-unreach)
PORT     STATE         SERVICE
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
1900/udp open|filtered upnp
5353/udp open          zeroconf
```

TCP portscan:

```
sabasahban@sabas-MacBook-Pro pythonProject5 %   sudo python Scanner.py --portscan --tcp 1 7000 -ip 127.0.0.1 127.0.0.1

Open TCP ports on 127.0.0.1: [1082, 3306, 5000, 5431, 7000]
```

```
sabasahban@sabas-MacBook-Pro pythonProject5 % sudo nmap -p 1-7000 127.0.0.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-29 01:30 +0330
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000024s latency).
Not shown: 6995 closed tcp ports (reset)
PORT     STATE SERVICE
1082/tcp open  amt-esd-prot
3306/tcp open  mysql
5000/tcp open  upnp
5431/tcp open  park-agent
7000/tcp open  afs3-fileserver
```