Author : Samarth Kamble.

CTF:VishwaCTF'23

Challenge Name : Phi-calculator

Category : Reversing

Level : Easy

Description : Python is a very easy and useful language for CTF players as well as for Hackers use some coding skills and get your flag.

Steps To Solve challange :

1. You will get a Phi-calculator.py python script.

2. If you run it you will see it is asking somw questions.

==========

Welcome to the Phi Calculator, vishwaCTF!

This is the trial version of Phi Calculator.

The full version may be purchased in person near

the galactic center of the Milky Way galaxy.

Available while supplies last!

==========

___Phi Calculator___

Menu:

(1) Estimate Projection Burn

(2) [LOCKED] Estimate Slingshot Approach Vector

(3) Enter License Key

(4) Exit Phi Calculator

What would you like to do, vishwaCTF (1/2/3/4)?

3. Check the python script carefully you will get some hints in TODO section.

```
#=========================================================================#
#============================Phi CALCULATOR===============================#
#=========================================================================#

import hashlib
from cryptography.fernet import Fernet
import base64



# GLOBALS --v
arcane_loop_trial = True
jump_into_full = False
full_version_code = ""

username_trial = "vishwaCTF"
bUsername_trial = b"vishwaCTF"
```

```
key_part_static1_trial = "vishwaCTF{m4k3_it_possibl3_"
key_part_dynamic1_trial = "xxxxxxxx"
key_part_static2_trial = "}"
key_full_template_trial = key_part_static1_trial + key_part_dynamic1_trial + key_part_static2_trial

star_db_trial = {
  "Sharuk Khan": 4.38,
  "Bollywood Star": 5.95,
  "Rohan 16": 6.57,
  "WISH 0855-0714": 7.17,
  "Tiger 007": 7.78,
  "Lalande 21185": 8.29,
  "UV Ceti": 8.58,
  "Sirius": 8.59,
  "Boss 154": 9.69,
  "Yin Sector CL-Y d127": 9.86,
  "Duamta": 9.88,
  "Ross 248": 10.37,
  "WISE 1506+7027": 10.52,
  "Epsilon Eridani": 10.52,
  "Lacaille 9352": 10.69,
  "Ross 128": 10.94,
  "EZ Aquarii": 11.10,
  "61 Cygni": 11.37,
  "Procyon": 11.41,
  "Struve 2398": 11.64,
  "Groombridge 34": 11.73,
  "Epsilon Indi": 11.80,
  "SPF-LF 1": 11.82,
```

```
  "Tau Ceti": 11.94,
  "YZ Ceti": 12.07,
  "WISE 0350-5658": 12.09,
  "Luyten's Star": 12.39,
  "Teegarden's Star": 12.43,
  "Kapteyn's Star": 12.76,
  "Talta": 12.83,
  "Lacaille 8760": 12.88
}



def intro_trial():
    print("\n================================================\n\
Welcome to the Phi Calculator, " + username_trial + "!\n")
    print("This is the trial version of Phi Calculator.")
    print("The full version may be purchased in person near\n\
the galactic center of the Milky Way galaxy. \n\
Available while supplies last!\n\
=====================================================\n\n")



def menu_trial():
    print("___Phi Calculator___\n\n\
Menu:\n\
(1) Estimate Projection Burn\n\
(2) [LOCKED] Estimate  Slingshot Approach Vector\n\
(3) Enter License Key\n\
(4) Exit Phi Calculator")
```

```
    choice = input("What would you like to do, "+ username_trial +" (1/2/3/4)? ")

    if not validate_choice(choice):
        print("\n\nInvalid choice!\n\n")
        return

    if choice == "1":
        estimate_burn()
    elif choice == "2":
        locked_estimate_vector()
    elif choice == "3":
        enter_license()
    elif choice == "4":
        global arcane_loop_trial
        arcane_loop_trial = False
        print("Bye!")
    else:
        print("That choice is not valid. Please enter a single, valid \
lowercase letter choice (1/2/3/4).")


def validate_choice(menu_choice):
    if menu_choice == "1" or \
       menu_choice == "2" or \
       menu_choice == "3" or \
       menu_choice == "4":
        return True
    else:
        return False
```

```
def estimate_burn():
  print("\n\nSOL is detected as your nearest star.")
  target_system = input("To which system do you want to travel? ")

  if target_system in star_db_trial:
     ly = star_db_trial[target_system]
     mana_cost_low = ly**2
     mana_cost_high = ly**3
     print("\n"+ target_system +" will cost between "+ str(mana_cost_low) \
+" and "+ str(mana_cost_high) +" stone(s) to project to\n\n")
  else:
     # TODO : could add option to list known stars
     print("\nStar not found.\n\n")



def locked_estimate_vector():
    print("\n\nYou must buy the full version of this software to use this \
feature!\n\n")



def enter_license():
    user_key = input("\nEnter your license key: ")
    user_key = user_key.strip()

    global bUsername_trial

    if check_key(user_key, bUsername_trial):
```

```
        decrypt_full_version(user_key)
    else:
        print("\nKey is NOT VALID. Check your data entry.\n\n")


def check_key(key, username_trial):

    global key_full_template_trial

    if len(key) != len(key_full_template_trial):
        return False
    else:
        # Check static base key part --v
        i = 0
        for c in key_part_static1_trial:
            if key[i] != c:
                return False

            i += 1

        # TODO : test performance on toolbox container
        # Check dynamic part --v
        if key[i] != hashlib.md5(username_trial).hexdigest()[7]:
            return False
        else:
            i += 1

        if key[i] != hashlib.md5(username_trial).hexdigest()[2]:
            return False
```

```
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[1]:
    return False
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[9]:
    return False
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[6]:
    return False
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[5]:
    return False
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[3]:
    return False
else:
    i += 1

if key[i] != hashlib.md5(username_trial).hexdigest()[8]:
```

```
            return False



        return True


def decrypt_full_version(key_str):

    key_md5 = md5.bmd5(key_str.encode())
    f = Fernet(key_md5)

    try:
        with open("keygame.py", "w") as fout:
            global full_version
            global full_version_code
            full_version_code = f.decrypt(full_version)
            fout.write(full_version_code.decode())
            global arcane_loop_trial
            arcane_loop_trial = False
            global jump_into_full
            jump_into_full = True
            print("\nFull version written to 'keygame.py'.\n\n"+ \
                  "Exiting trial version...")
    except FileExistsError:
            sys.stderr.write("Full version of keygame NOT written to disk, "+ \
                             "ERROR: 'keygame.py' file already exists.\n\n"+ \
                                      "ADVICE: If this existing file is not valid, "+ \
                                      "you may try deleting it and entering the "+ \
```

"license key again. Good luck")

def ui_flow():

intro_trial()

while arcane_loop_trial:

menu_trial()

Encrypted blob of full version

full_version = \

b"""

tNNNNNOtG_aiUNjCnJny_64Tvhoso7V87ZY4NAc4t-
rHoZGdfp4nfJltacKpnW5SYnuKKQphy9kCQdVCClgvM9nZz25F6qtsv4MCiZ5VHFaaAwx6qkLNXfK5Lq72OI4
REedqAAa2wMecumyI4n4tL-
KI_0MUbiSyUuRcCDaGgT_5EGRGVq0kNQ5X5vNpgxV9n3C4fk6RkOD082RhCSyaJgHTy0qfRQUure3kG_yM
r9WC5HNpBWfbP9NW7A3L1XwJKNGmnOxKj6KGamdQUh9Lpssj-v-TsDC-
16uS_s2JOR9aDdavSh6GUAhNidjt0KOafsiI0Sb3ZGcBA6UcrV3rVKdq4gYgfsuApCn99htehps0y19M5rSiegZZ
tzsJ_9ytiB7HpPsg79FuiDJRUwuIrvQXOMb7GtGW-1jcO92boU_oSTWcZpfc1j42gQRWzniaEFXKy39cu9-
ptIKHXGsfwHoWPtgMsE8lw28WSPqzRGh2xxg_qJ4nYA8OGrEUYHcPRbq9kOHSkmDWMAkrl6VFa2w-CGE-
lkPKeP2_N3GPOpdjHWLivC6rzYXCFOEWO8qxEyJzlyaZU4nLq6LKCaL457gx6HcTB6Rmj4X4QRuSgZFB4Id2H
uNF85w8xbxp9_TT2i315hdIM-GL7aH7kxufegRSX0u-
0wvvYoGXYiBo3mcKd0RYqK9_JRX681YfVhSReufiiCzzKk1X7VQyzwVJxLj--7ydcKCIey9YnyV-
7acBS4ZLrg3wyU9i5L83X3IQPeAMwU8hdX4cGXb0_V4UBzRgsr8ctuNLQyqzD8jicuUEu4HRZ34DptPWn3IU
1KNh2ZEQjoRJpaKkhzg_kY2jKOGSNCMJkeavbEhamj5UaedJ6Amv871KvW0BUDmg_hyitkQzSZkNvFcmz9L
WbkfcGT1ucFdYr5VHVPOKRubstGNuUrCTss-Dv20eQLDZDvb8mblI2MCEwXIx8LQTMquhFDnXYk-
QEqiXOmzNLdwoizoP_4Gg3_nzKydkrYEYN9LC7igki9L65JM-
8QrTqMGvaHtwu6kdWU0kWisRuKVGBRvSTMfrK2xCCT4cK1aQPM8E9xftUkxcaJ24fDFITzk5QC0kTvuszNO
p98ont-def-DVo7LdjWdX0-
0A0g7uSXS671qbK07KJpVTYWhSM0ZUkCBVwIVJA8Ko7zXWvY7tbDU8khl9SpR6j8_TIj5A9asJHPSqMXRAL
W5JL3vK20BgWtvLGjiPGrgs-
jQJw_SU6m_uchsV1fQu9yB9RaNukcbAb4wZwP3rMHXCxxHs_tsiwJzaN89Tifhkw70ymjM1650vfv0_WCgQ
VJXnxfcemVJ8LA-
ZrhOLl_s3WWBgH4pPt5fVaGZ3LJ5ThcWZB4u6L6ix1DCkJLZ5Ae5_pO7v2SFlg7QL72Y_rqr8LAWkpEPOxs3r
Q-
3nB6XzCNoos_48nnZ3Y1nkAIXjhoJ9Zrp6LEbI0WjtWZ0Xz3LNTY_logLK5WrbCDmbBDOj_Ihr2x8CfaoB6a2c
3npnL8L-6MuXaeFOnrhNPFMgGdWG4_JKLfyDlK-Cty-
ywpd84U0NNCAaW9NyAMjiTY6sXoqpkcpD7EA8sqbH6oW2d2KrpKerq2KsR2HUX-DGnpz-

nzS5Zg4JeAyS8ETEhPntal7b6KlRLB_-klbjcHLfBN9p5w6h8dcwh0qbauqe0BJJXUcVJiBfQqmK-
LRpDisqKsYqQYQFQdTWHlO5tvDX3VIdHrONA2MUqXlSNNPHbt4H0EWYW3gRrqS_CYM5rdUlb_wjsOzdEv
_oX2pjhLH_cfkGOO6b2n2b1ik-
aceC4DSIkqJQ7ol4IGSXPIJ2lNTxY1BUCNp6upbIulfNVZDuudWS0FKKqdrKmSeZ7crke3fI9hY_E_PpxaBx28IR
0VleiWYZw1fV-ZxKdESGqjwhzc6sarDvmOUNl2Xx7TtH7WYjFitHITOF581VGkhD-
wrMJ5bq1z9m6kLYZXKAFI-RHMKuTCBm4xq87tGEkQZq9F5cxFdsvODteVeRhAgnQLWfp22e5ZNTAcr-
bhTuR_DEZCQnRIC8PvoAh0jawteg_4Ds8Z6MHEl6smffOfdVwsSlzWQr8hFzm5lbfiGhEsfwpP_zulrISewvUF
mU5BRsAF0vuCZV6U9w4iqvq0Jx3rjwsG3ecfnqohUOWGEkCpOA-
qp8inYiYmpWrulTDuiRIjzvlpwNW16ctBG5-eE0-
MYXbvnnn3BUpf8EO3MKYr9YxWUfdPiwTrV4ddyxYsNuT08tAfNkgpoLNRImXQiAQCqoJBfvbVK3yXXvvTAm
gMZehGuZjlpHD1mEA_5fEP5QGDQi0y3xn0BRYJ2H04Bt5Fw5k1h2eqJFOn9aRV6YW7aac-cYGTb-
P7gfd1obGm4JqUAZiNC2TJi99ASA37cn8HL6zwzqZNt0Ccj9eskrTXd60wu-
IpOhL3Lih1t2_Agi1r8PrX1wAKy06mYTYOB35uYjvk4HpDzH_9Z0i1DfLswLOEJ5fHawpO3yTS8XWt6CLToUpi
NRQydj12MgVGSBnVduxciFmosT1YDS7r_AsuKvwtOZuWOht9gGnli0t5H2PChM-
O4m_FxzRiA_rH5T1eug3Mi0ltGhBKJ0Vvt0KkSB02DhtMFdVEt7sTEw6skQIJKiDG6x3mKyKkA6YeULUopJ7V
ef0cYkz7csNCOLyaSjEGUKV4UuaZHfCvX3i0bCH7VO67l4KPHZapZpTfgEO4mdanMV7qE8LCDsNMD0PHwG
5M_U52Dc9rx5U_T48io0QSP1dmtcAysUpKeOYhus_Tpp_1qma9R4MjbvS16nWuUCUFNTuBjpyYl7kkl22M
raMIrXKpKeN7wHoCTpF-
FJzHwS1VCr_Cxcsptv5eVnkHuPJK5wX0p_a0_d2HNi9XNXWOnJfgwpLOkghHgUGSWQ0xl9IQBdIWk1-
I6gQ1yAfaS1SsAesVcO9LxbPkEVKQhOvPFntvjn830F6-1oERMZMht-
rgmwe1Js7pB1CGQq61WFZ252QJdUIIYDf8lLXzumfMksrV_hI5T7L8sijVLOO4xeSEwcSE4-sTjS4Izn-
kMye6l9mvVYAHldm2h4SOzZwp1I8LtrddKfYVhFUS6TQuiTKd-zRdYGJakFNR-
T_mrK7dCQNyfFi_qEYOlD0MrxORD1LoPczaoMVCGW_VlMYK0MOBm3bp0wh5zSHSNmA8fWyjhMSU2TD
rP9G2TWB8yWRua4AdvhqmzKIZreqEnY1P9MoWsTFRxhRXDY2V2ArJ5Az7q4ZFgUqgMuB190_yKC2CD8G
hm5OeCyLXtTs76AMfuNH0KXKtylGJDXmBAIi6251du4jcZtJJSz8In_mTyKASq8DzDJcouxJGYzb9vkV4J92uxj4
burWIR5a9YO1UJm50bFnwI_2wWJ_5Oq5Tgm6F3D2K_ksN_GtElrG0QKtoD8zLk_A_43F_Q94hq66-
AaEN_N1XT-hh31XU5ogHt6s3-bkbB4jnCJ8-uZ0neAyTwERt0_YuNZNYxausWyab2IaDb2RktKw6i-
xnOyGhu4wg5iourcQ5RtgTioKG4zlcDoF49YN3FkPkRd7iQFkUsaYXJV84VyNrH8ADR6qeDq9VTD9yEJMQm
Utim7qB6Bt4cVg7D6HN2ARVp6MAQGftugXSIrc19q7aTWQg-4-HPSWFUJOuGXrdo_N34KB4G5H7k-
PKdcufOjVqZbCKUeJkubSLnC6yCWIBelm8GRLQYfUoIqzuLWgN0oCtZCP1eAV-
FdplHdiMSTcVWjQpTtuGGF1h8KwZyExkBkhRZQB364NqYgehfyxKwcq2AhHOHSjAJoDosLVP5zrCdkp_Cupn
IkZKULeSu2PYdKK7HupMkUG9P8ED==

"""

Enter main loop

ui_flow()

if jump_into_full:

```
    exec(full_version_code)
```

4. Try to decode it and write another python script using help of this python script .

5. Here is the pyhton script to solve this quesion .

```
import hashlib

username_trial = "vishwaCTF"
bUsername_trial = b"vishwaCTF"

res = ""
key_part_static1_trial = "VishwaCTF{m4k3_it_possibl3_"
key_part_dynamic1_trial = "xxxxxxxx"
key_part_static2_trial = "}"

#I used bUsername_trial because enter_liscence used it as well but after testing afterwards, they output the same answer
res += hashlib.md5(bUsername_trial).hexdigest()[7]
print(res)
res += hashlib.md5(bUsername_trial).hexdigest()[2]
res += hashlib.md5(bUsername_trial).hexdigest()[1]
res += hashlib.md5(bUsername_trial).hexdigest()[9]
res += hashlib.md5(bUsername_trial).hexdigest()[6]
res += hashlib.md5(bUsername_trial).hexdigest()[5]
res += hashlib.md5(bUsername_trial).hexdigest()[3]
```

```
res += hashlib.md5(bUsername_trial).hexdigest()[8]
print(res)

key_part_dynamic1_trial = res
key_full_template_trial = key_part_static1_trial + key_part_dynamic1_trial + key_part_static2_trial

print(key_full_template_trial)
```

6. After runing this python script you will get your flag

7. Flag : VishwaCTF{m4k3_it_possibl3_e47e1fbf}